

https://dspace.org/

A Comparative Performance Analysis of Popular Deep Learning Models and Segment Anything Model SAM for River Water Segmentation in CloseRange Remote Sensing Imagery

Armin Moghimi; Mario Welzel; Turgay Celik; Torsten Schlurmann

https://hdl.handle.net/10539/42881
WOS

Downloaded from a DSpace repository. https://dspace.org/

IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 19 February 2024, accepted 31 March 2024, date of publication 5 April 2024, date of current version 17 April 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3385425

RESEARCH ARTICLE

A Comparative Performance Analysis of Popular Deep Learning Models and Segment Anything Model (SAM) for River Water Segmentation in Close-Range Remote Sensing Imagery

ARMIN MOGHIMI[1], MARIO WELZEL[1], TURGAY CELIK[2,3,4], AND TORSTEN SCHLURMANN[1]

[1]Ludwig-Franzius-Institute for Hydraulic, Estuarine and Coastal Engineering, Leibniz University Hannover, 30167 Hanover, Germany
[2]School of Electrical and Information Engineering, University of the Witwatersrand Johannesburg, Johannesburg 2050, South Africa
[3]Wits Institute of Data Science, University of the Witwatersrand Johannesburg, Johannesburg 2050, South Africa
[4]Faculty of Engineering and Science, University of Agder, 4630 Kristiansand, Norway

Corresponding author: Armin Moghimi (moghimi@lufi.uni-hannover.de)

This work was supported in part by the joint research project "Zukunftslabor Wasser" funded by the Lower-Saxon Ministry of Research and Culture under Grant FKZ: 11-76251-1873/2022 (ZN3994), and in part by the Open Access Fund of Leibniz Universität Hannover.

ABSTRACT Accurate segmentation of river water in close-range Remote Sensing (RS) images is vital for efficient environmental monitoring and management. However, this task poses significant difficulties due to the dynamic nature of water, which exhibits varying colors and textures reflecting the sky and surrounding structures along the riverbanks. This study addresses these complexities by evaluating and comparing several well-known deep-learning (DL) techniques on four river scene datasets. To achieve this, we fine-tuned the recently introduced "Segment Anything Model" (SAM) along with popular DL segmentation models such as U-Net, DeepLabV3+, LinkNet, PSPNet, and PAN, all using ResNet50 pre-trained on ImageNet as a backbone. Experimental results highlight the diverse performances of these models in river water segmentation. Notably, fine-tuned SAM demonstrates superior performance, followed by U-Net$_{(\text{ResNet50})}$, despite their higher computational costs. In contrast, PSPNet$_{(\text{ResNet50})}$, while less effective, proves to be the most efficient in terms of execution time. In addition to these findings, we introduce a novel river water segmentation dataset, LuFI-RiverSnap.*v*1 (`Dataset link`), characterized by a more diverse range of scenes and accurate masks compared to existing datasets. To facilitate reproducible research in remote sensing and computer vision, we release the implementations of the fine-tuned SAM model (`Code link`). The findings from this research, coupled with the presented dataset and the accuracy achieved by fine-tuned SAM segmentation, can support tracking river changes, understanding river water level trends, and exploring river ecosystem dynamics. These can also provide valuable insights for practitioners and researchers seeking models tailored to specific image characteristics with practical means in disaster risk reduction, such as rapid assessments of inundations during floods or automatic extractions of gauge data in watersheds.

INDEX TERMS Deep learning, segment anything model (SAM), river water segmentation, U-Net, DeepLabV3+, LinkNet, PSPNet, PAN, RiverSnap.

I. INTRODUCTION

Rivers, as vital components of Earth's hydrological cycle, play a fundamental role in sustaining ecosystems and human societies [1]. Accurate delineation and characterization of river boundaries and features are imperative for a multitude of applications, including flood risk assessment, water resource management, riverine hydrological parameter estimation, and ecological conservation [2]. In contemporary river research studies, where high-resolution images with exceptional clarity and detail are essential, e.g. either to validate numerical models or to acquire accurate data for

The associate editor coordinating the review of this manuscript and approving it for publication was Derek Abbott.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

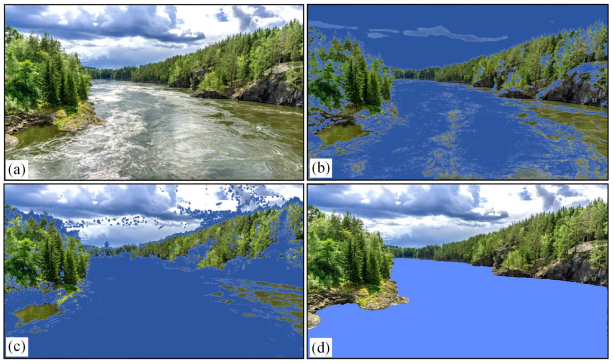


FIGURE 1. Examples of different segmentation models in river water segmentation. (a) The image of Norway's Glomma river (source link) and its segmentation resulted from (b) image thresholding, (c) hybrid algorithm (k-means + active contour model), and (d) Deep Learning (DL) model.

early warning systems, the segmentation of river water in close-range Remote Sensing (RS) images captured by low-cost sensors becomes particularly significant [3], [4]. Indeed, close-range RS images captured by low-cost cameras (e.g., smartphone/surveillance cameras) are proven to facilitate the detection of subtle variations in river water properties and the surrounding terrain [3], [5], [6]. This presents, until yet, a rarely utilized and systematically investigated opportunity to extract nuanced insights into hydrological parameters or any related process (e.g., water level, water turbidity, floating debris, etc.) from close-range RS images [3], [6].

Segmenting the water body from the background in close-range images is a key step in determining riverine parameters. It forms the basis for further analysis, directly impacting the accuracy of subsequently analyzed parameters through images. In developing these methods, researchers have introduced a variety of image segmentation algorithms, classifying them into conventional image analysis and advanced Deep Learning (DL) models [7], [8]. While traditional methods (e.g., thresholding, region-based, and hybrid algorithms) are commonly used for image segmentation, they encounter limitations when applied to water bodies like wetlands, lakes and river scenes due to the complex nature (e.g., inhomogeneous appearance and color variations), and the reflections of surrounding structures (e.g., vegetation, rocks, and buildings) and the sky on the water surface [9], [10], [11] (see Fig. 1 (b) and (c)). This is mainly because most of these techniques depend on low-level/basic image features for object segmentation, which may not inherently capture complex spatial relationships, such as those found in river scenes in close-range images [8], [12]. In contrast, DL models, known for their automatic extraction of high-level semantic features from various data types, have recently offered more robust solutions in this specific aspect [5], [11], [12]. Furthermore, since DL models are typically trained on diverse datasets, they exhibit superior capabilities in handling sky reflections and structures on the water body, as depicted in Fig. 1 (d).

While DL techniques are mainly applied to segment water in space/airborne RS images [13], [14], [15], they have high potential in accurately extracting hydrological features such as water bodies and water levels from close-range imagery captured by low-cost cameras, such as surveillance and smartphone cameras [12]. For example, to predict floods stemming from river overflow, Lopez-Fuentes et al. [16] identified rising river water levels by applying three DL techniques (Fully convolutional network (FCN) [17], DenseNet [18], and [19]) through water segmentation in surveillance camera images. Their study showed that the DenseNet model exhibited superior performance in flood extent detection from close-range RS images. Pan et al. [20] also compared different image-based methods for water-body segmentation in surveillance camera images to create a real-time water level detection service. The study revealed that employing a convolutional neural network (CNN) at the core of the service significantly enhanced accuracy, yielding highly accurate results (about 9 mm error) compared to reference measurements. Vitry et al. [21] employed U-Net [22] to detect floodwater and introduced the Static Observer Flooding Index (SOFI) to obtain water level fluctuations. In a similar approach, Akiyama et al. [23] examined the potential of the SegNet [24] segmentation to extract river water from the background of close-range RS images.

Vandaele et al. [25] employed both the UperNet [26] and DeepLabv3 [27] networks with a ResNet50 architecture [28] as a backbone for the river water segmentation process in the context of water level detection. Muhadi et al. [12] successfully utilized DeepLabv3+ [29] and SegNet networks to detect water bodies, evaluate water levels, and track fluctuations in surveillance images. Eltner et al. [11] also integrated advanced DL water segmentation models (SegNet and FCN) with photogrammetric techniques to achieve precise water stage measurements from images taken by a Raspberry Pi camera.

The aforementioned studies have yielded valuable insights into identifying water bodies in image backgrounds and their application in extracting hydrological parameters, such as water level. However, most of these studies relied on a limited set of segmentation models, leaving their adaptability across diverse environmental contexts and image datasets uncertain. Furthermore, their applicability was often confined to specific scenes (e.g., datasets used in [11]), making the trained CNNs less suitable for broader applications in different environments [3]. To enhance real-time water level monitoring and address these limitations, Eltner et al. [30] introduced UPerNet [26] with the ResNeXt-50 backbone as a well-generalized CNN model, chosen from a range of DL models for water segmentation in various geographical contexts. Furthermore, Wagner et al. [3] conducted a widespread study evaluating 32 DL segmentation models by introducing high-quality RIWA.*v*1 [6], [31] for water detection with online/offline augmentation.

While existing studies have identified several effective DL models for river water segmentation, limited datasets often

constrained their evaluations. A more comprehensive analysis across multiple datasets with diverse conditions is essential to better assess the capabilities and generalizability of each model. While the RIWA dataset [31] provides valuable support for this research area, there remains a critical need for additional accurately labeled datasets to further enhance model training and evaluation.

Moreover, DL models such as U-Net [22], DeepLabV3+ [29], Pyramid Scene Parsing Network (PSPNet) [32], have been widely employed in river water segmentation and its applications. For instance, a recent comparative analysis by [3] revealed that U-Net with a ResNeXt50 backbone had the best performance for river water segmentation among 32 DL models. However, while this study sheds light on the capabilities of established DL architectures, it also raises questions about the exploration of alternative methodologies and newer model designs. In particular, the potential of efficient model architectures like the Pyramid Attention Network (PAN) [33] and LinkNet [34] as well as recently published DL models, such as Meta AI's "Segment Anything Model" (SAM) [35] have not yet been explored and compared. Incorporating such advanced models could also unveil new avenues for research in river water segmentation.

In this study, we aim to address these challenges, making the following contributions:

1. For the first time, to the best of our knowledge, we fine-tuned the SAM segmentation model for water/river water segmentation in close-range RS images. This fine-tuning aims to enhance the segmentation of water bodies, particularly for wetlands, lakes, and rivers. The fine-tuned SAM model's code has also been made accessible at: (`Fine-tune SAM code link`).
2. We extensively surveyed six state-of-the-art DL models (U-Net [22], DeepLabV3+ [29], LinkNet [34], PSPNet [32], PAN [33], and the recently published SAM [35]), evaluating their strengths and limitations. This assessment aims to establish their suitability for water body segmentation from close-range RS images at rivers.
3. We collected close-range images from rivers using various platforms, such as smartphones, cameras, surveillance cameras, and low-altitude drones, to introduce a new dataset named LuFI-RiverSnap.*v*1. It comprises over 800 images with precise annotations, along with a few suitable images from the Kaggle WaterNet [36], Elbersdorf / Wesenitz [11], RIWA.*v*1 [31] datasets. To support and advance the development of river water segmentation tasks, we have released the recently introduced LuFI-RiverSnap*v*1 dataset at (`Dataset link`).
4. We performed a thorough experimental analysis of segmentation methods on three considered benchmark datasets and the newly introduced LuFI-RiverSnap.*v*1 datasets. This comprehensive evaluation sets a research baseline, providing valuable segmentation model recommendations for river water analysis and suggesting future study directions.

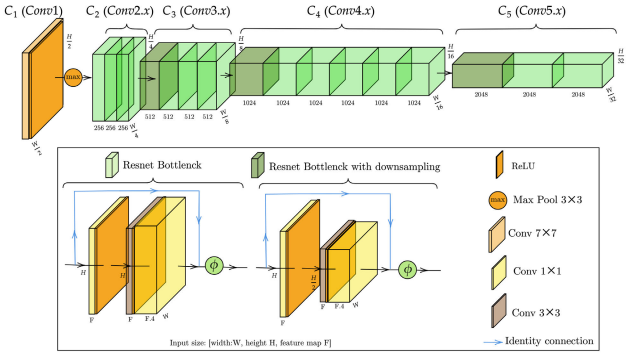


FIGURE 2. The ResNet50 backbone adopted from the PyTorch implementation and [38].

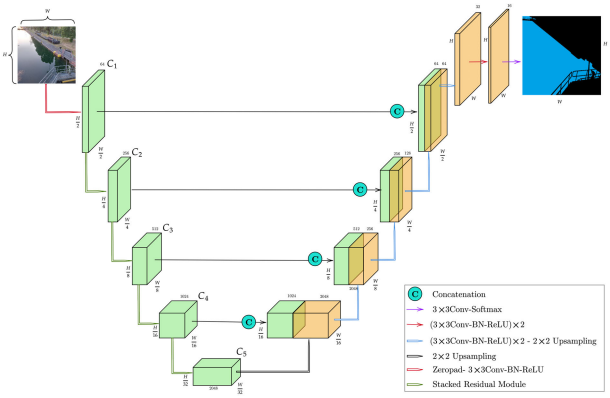


FIGURE 3. Architecture of U-Net$_{(ResNet50)}$, adopted from [39].

II. MATERIALS AND METHODS

A. DL SEGMENTATION MODELS

This section briefly overviews the unique architectural features of the six baseline DL models considered in the context of water body segmentation tasks. For all models except the SAM, we employed ResNet50 [28] (see Fig. 2), pre-trained on the ImageNet dataset [37], as the backbone architecture.

1) U-Net

U-Net, introduced by Ronneberger et al. [22], is widely used for object segmentation. Its U-shaped architecture includes an encoder for downsampling and a decoder for upsampling and feature fusion [22]. Skip connections aid in integrating features from different resolutions [39]. ResNet50 serves as the encoder in this study, while a custom decoder involves upsampling and convolutional layers. (see Fig. 3).

2) DeepLabV3+

DeepLabV3+ is an advanced variant of the DeepLab series, featuring Atrous Spatial Pyramid Pooling (ASPP) and an encoder-decoder structure [29]. ResNet50 is also integrated into the encoder pathway in this study. It begins with extracting high-level features, and comprehensive representation through feature combination. Upsampling with transposed convolutions enhances spatial resolution, enriched by skip connections (see Fig. 4),

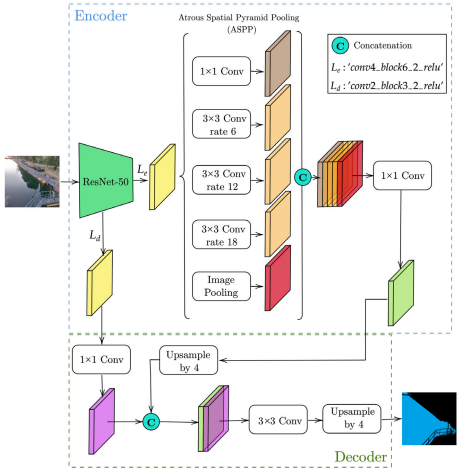


FIGURE 4. Architecture of DeepLabV3+$_{(ResNet50)}$, adopted from [29].

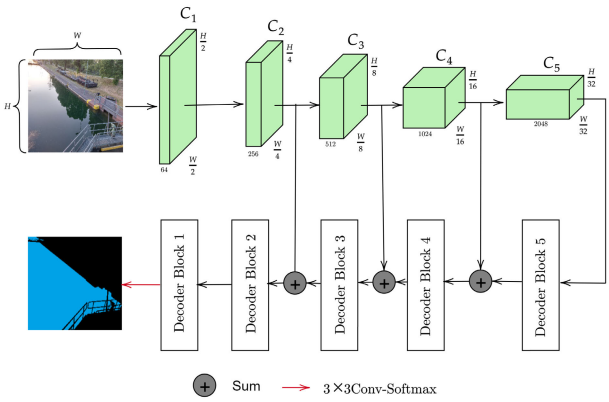


FIGURE 5. Architecture of LinkNet$_{(ResNet50)}$, adopted from [34].

3) LinkNet

LinkNet features a U-shaped architecture, maintaining the crucial encoder-decoder structure for hierarchical feature extraction and seamless upsampling [34]. In the depicted architecture Fig. 5, ResNet50 is also utilized in the encoder to extract high-level features through a downsampling strategy. Encoder features are directly linked to corresponding outputs in the decoder via skip connections using the ''sum'' operator.

4) PSPNet

PSPNet is a model structured on the encoder-decoder paradigm, incorporating a distinctive pyramid pooling module (PPM) for aggregating contextual information across multiple scales within the feature hierarchy [32]. As shown in Fig. 6, the initial step involves using ResNet50 to extract the feature maps. Following this, PPM is employed to generate representations of different sub-regions. These features are then upsampled and concatenated, enabling the model to incorporate both local and global contextual data. The final representation undergoes convolutional processing to achieve river water predictions.

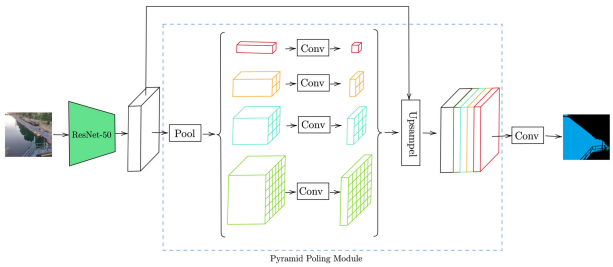


FIGURE 6. Architecture of PSPNet$_{(ResNet50)}$, adopted from [32].

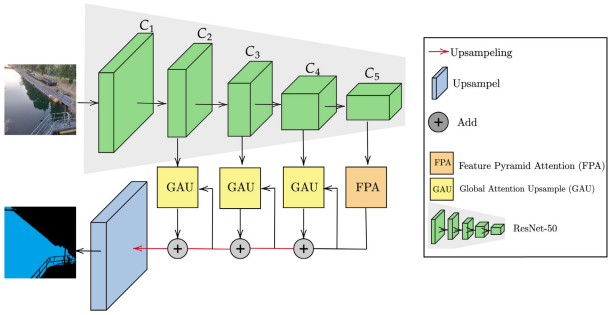


FIGURE 7. Architecture of PAN$_{(ResNet50)}$, adopted from [33].

5) PAN

s an encoder-decoder architecture to enhance global contextual information in semantic segmentation [33]. This is achieved by integrating the Feature Pyramid Attention (FPA) and Global Attention Upsample (GAU) modules. In the implementation, the ResNet50 was also used to extract dense features, followed by FPA and GAU for accurate pixel predictions and localization details (see Fig. 7),

6) SAM

SAM stands as an innovative encoder-decoder promptable model developed by the Meta AI team for precise image segmentation [35]. Its training involved a massive SA-1B dataset, encompassing over 1 billion masks extracted from 11 million images, raising its generality to segment unseen objects and images [35], [40]. This prowess extends the model's applicability beyond the confines of image segmentation, allowing it to be effectively employed in various scenarios, including but not limited to object tracking [40].

As depicted in Fig. 8 (a), the SAM comprises three key components [35]. First, an image encoder denoted by Enc_I uses a Vision Transformer (ViT) backbone, such as ViT-B (91M), ViT-L (308M), or ViT-H (636M parameters), to process 1024×1024 images I and generate image embedding features F_I [35]. The flexible prompt encoder, denoted by Enc_p, then adeptly handles both sparse prompts P_s (e.g., points, boxes, text) and dense prompts (masks) P_l, translating them into tokens T_p and T_L, respectively [35]. Finally, the outputs of the encoders pass to a lightweight mask decoder Dec_L [35] for label predictions.

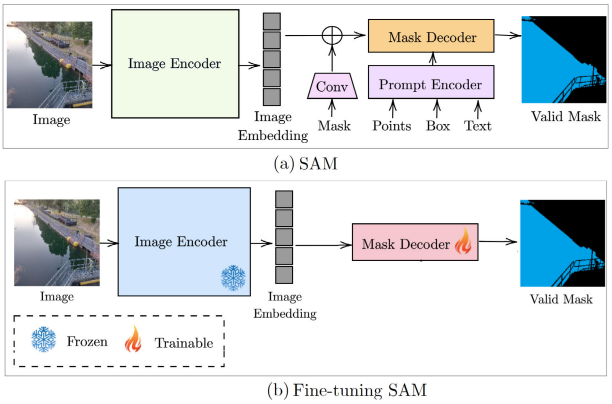


FIGURE 8. Architecture of (a) SAM, adopted from [35], and (b) fine-tuning of SAM for river water segmentation in this study.

7) FINE-TUNING OF SAM

In this study, SAM was fine-tuned for river water segmentation, as depicted in Fig. 8 (b). In this way, We first froze the SAM encoder Enc_I with a lighter ViT-B (91M parameters) backbone to optimize computational efficiency and conserve resources. This configuration allowed us to use the pre-trained parameters of the SAM encoder for extracting robust image embedding features F_I from river scene images. This also facilitates the use of larger batch sizes, thereby enhancing the overall efficiency of the model training pipeline.

Given the absence of a prompt in our fine-tuning process, the learnable mask tokens T_L were automatically obtained through element-wise addition of a learned embedding to each location of the image embedding F_I [35]. Subsequently, the extracted features F_I alongside the learnable mask tokens T_L were directly fed into the trainable mask decoder Dec_L to predict the low-resolution mask $\hat{L}$ as follows:

$$\hat{L} = \text{Dec}_L(F_I, T_L) \tag{1}$$

The low-resolution mask $\hat{L}$ is then upsampled to the input size and compared with the ground truth mask using a determined loss function during training.

Our fine-tuning process is conducted without any prompt encoder, enabling a fair assessment of SAM alongside other models under consideration and reducing human-machine interaction.

B. RIVER WATER SEGMENTATION USING DL MODELS

In this study, the river water segmentation workflow using DL models is depicted in Fig. 9. Supported by a river water dataset with distinct subsets for training, validation, and testing, this methodology forms the foundation of our approach.

Let $\mathcal{D} = \{(I_i, L_i)\}_{i=0}^{N-1}$ denote our river dataset, where $I_i \in \mathbb{R}^{w\times h\times 3}$ is an RGB image of the river scene, and $L_i \in [0,1]^{w\times h}$ is its corresponding annotated river mask for sample i, i.e., "0" represents the background, and "1" denotes the river. The dataset D includes three mutual subsets: the training set ($\mathcal{D}_{\text{train}}$), the validation set ($\mathcal{D}_{\text{val}}$), and the test set ($\mathcal{D}_{\text{test}}$), where $\mathcal{D} = \mathcal{D}_{train} \cup \mathcal{D}_{val} \cup \mathcal{D}_{\text{test}}$. Initially, all subsets are subjected to a preprocessing step, denoted as P, which includes normalization ($\mathcal{N}$) and resizing ($\mathcal{R}$) operations:

$$\mathcal{D}^P = \{(\mathcal{R}(\mathcal{N}(I_i)), \mathcal{R}(L_i))\}_{i=0}^{N-1} = \{(I_i^P, L_i^P)\}_{i=0}^{N-1} \tag{2}$$

where, $\mathcal{N}(.)$ ensures consistent pixel values, and $\mathcal{R}$ standardizes the input dimensions to a 512×512 pixel format, resulting in $I_i^P \in \mathbb{R}^{512\times512\times3}$ and $L_i^P \in [0,1]^{512\times512}$ as the preprocessed RGB image and the intended annotated river mask for sample i, respectively. Subsequently, preprocessed training and validation samples are fed into a chosen deep learning segmentation model represented by f_M, where M can be one of the segmentation models from (SAM, U-Net$_{(\text{ResNet50})}$, PSPNet$_{(\text{ResNet50})}$, PAN$_{(\text{ResNet50})}$, LinkNet$_{(\text{ResNet50})}$, DeepLabV3+$_{(\text{ResNet50})}$), with parameters M. During the training phase, the model f_M is constructed and refined using the dataset $\mathcal{D}_{train}^P$, guiding it to learn river water patterns and handle reflection structures. To ensure the model generalizes well to new data and avoids overfitting, the dataset $\mathcal{D}_{\text{val}P}^P$ is also utilized during training.

Given the preprocessed input image I_i^P, the model computes the predicted binary river mask $\hat{L}_i^P = f_M(I_i^P; \theta_M)$ through a forward pass in each training loop. To quantify the difference between the predicted water mask $\hat{L}^P$ and its ground truth mask L^P over each training loop, the loss function is computed using binary cross-entropy (BCE) with sigmoid loss:

$$\mathcal{L}_{\text{BCE}}(\hat{L}_i^P, L_i^P) = L_i^P \log(\sigma(\hat{L}_i)) + (1 - L_i^P)\log(1 - \sigma(\hat{L}_i)) \tag{3}$$

where, σ is the logistic sigmoid function. To update the model parameters θ_M, the L_{train} is optimized using the *Adam* optimizer [41] over each epoch. Moreover, the validation loss L_{val} is computed over the $\mathcal{D}_{\text{val}}^P$ using the trained model f_M and its updated parameters θ_M, to monitor the model's performance at the end of each epoch. Algorithm 1 outlines the specific steps involved in the training phase of river water segmentation.

After fine-tuning the model, evaluation is performed on the test samples. In this way, the trained model f_M and its optimized parameters θ_M are employed to predict the test subset $\mathcal{D}_{\text{test}}^P$. These predictions are further compared with the ground truth masks L_{test}^P using an array of metrics detailed in [subsection II-D] to evaluate model performance. The specifics of the testing process have been summarized in Algorithm 2.

C. DATA

We employed three established benchmark datasets (i.e., Kaggle WaterNet [36], Elbersdorf/Wesenitz [11], RIWA.*v*1 [31]) along with our newly proposed LuFI-RiverSnap.*v*1 dataset to enrich the comprehensiveness of our analysis through various baseline deep learning networks. The amount of training, validation, and testing images of these datasets is shown in Table 1.

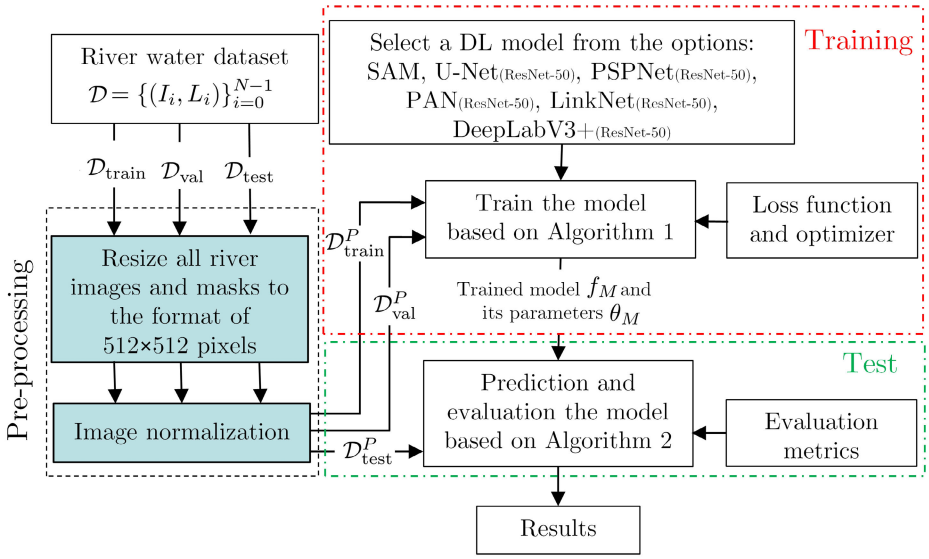


FIGURE 9. Workflow of river water segmentation using Deep Learning (DL) models.

1) KAGGLE WaterNet [36]

This widely adopted public dataset, available in two versions, contains over 1000 annotated images (`Dataset link`). It integrates water images from the *ADE20K* dataset, encompassing diverse water body scenes such as water streams, river segmentation, and gauges from various locations. In this study, we selected 748 accurately labeled river scene images from the second version. This dataset was divided into training (70%), validation (∼16%), and test (∼14%) sets, forming the basis of our workflow. Despite the moderate range of diversity in river water color and reflection, most considered models are expected to exhibit proficiency in the river water segmentation task on this dataset.

2) ELBERSDORF/WESENITZ [11]

This dataset comprises 3,407 accurately annotated images captured using a Raspberry Pi RGB camera positioned 4 meters above the Wesenitz river at the Elbersdorf station between March 30, 2017, and April 30, 2018 (`Dataset link`). We randomly selected 1001 images from this dataset, allocating approximately 50% for training, ∼20% for validation, and 30% for test samples. Given that this data originates from a single location, optimal model performance is anticipated on this dataset.

3) RIWA.*v*1 [31]

This stands as a robust and versatile resource for river scene analysis and segmentation, encompassing 1128 images captured using smartphones, Unoccupied Aerial Vehicles (UAVs), and DSLRs (`Dataset link`). This dataset incorporates diverse river scenes, including images from Elbersdorf/Wesenitz [11] and Kaggle WaterNet [36]. The dataset's diversity, with rivers of varying colors, presents a challenge for models in segmenting water from the background.

4) LuFI-RiverSnap.*v*1

This dataset includes close-range river scene images obtained from various devices, such as UAVs, surveillance cameras, smartphones, and handheld cameras with sizes up to 4624 × 3468 pixels (`Dataset link`). Several social media images, as typically volunteered geographic information (VGI) [42], have also been incorporated into the dataset to have more diverse river landscapes from various locations and sources. The images mainly include river scenes from several cities in Germany (Hannover, Hamburg, Bremen, Berlin, and Dresden), Italy (Venice), Iran (Ahvaz), the USA, and Australia. To further enhance the dataset's diversity and accuracy, a small subset of images of Elbersdorf/Wesenitz [11], RIWA.*v*1 [31], and Kaggle WaterNet [36] have been added to the data. This comprehensive dataset includes 1092 images, all accurately annotated, establishing it as a valuable resource for advancing research and development in river scene analysis and segmentation. The dataset comprises challenging cases for water segmentation, such as rivers with significant reflection, shadows, various colors, and flooded areas. Fig. 10 provides an insightful overview of the LuFI-RiverSnap.*v*1 dataset, showcasing instances of rivers with different colors.

D. EVALUATION CRITERIA

To evaluate the performance of considered DL models in river water segmentation, their predictions were compared with the corresponding ground-truths of the test samples using several metrics extracted from the confusion matrix. These metrics include the Overall Accuracy (*OA*), *Precision*, *Recall*, F1-score (F_S), Intersection over Union (*IoU*), and Kappa Coefficient (κ), measured by:

$$OA = 1 - \frac{TP + TN}{TP + FP + FN + TN} \times 100\% \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \times 100\% \tag{5}$$

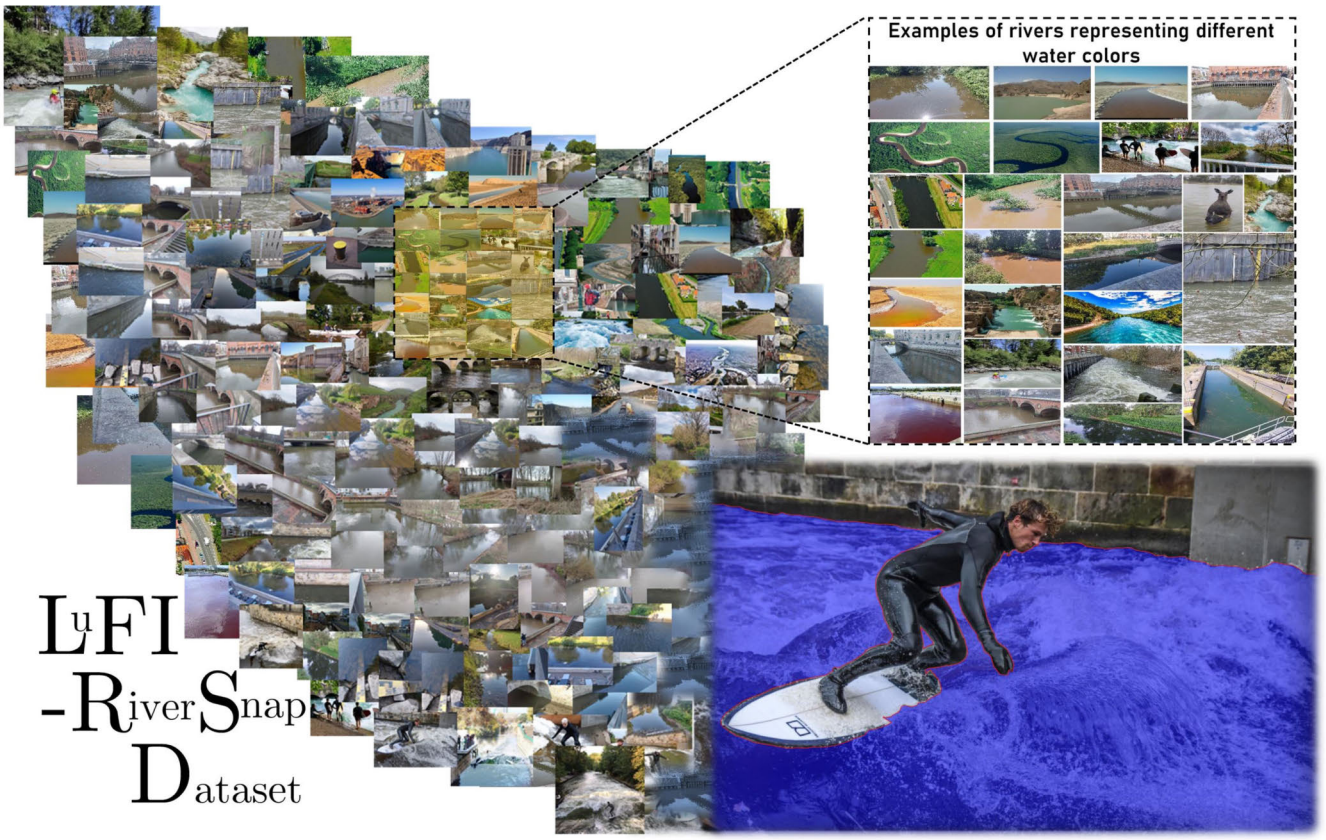


FIGURE 10. LuFI-RiverSnap dataset- Illustrative examples of rivers featuring diverse water colors.

$$Recall = \frac{TP}{TP + FN} \times 100\% \tag{6}$$

$$F_S = \frac{2TP}{2TP + FP + FN} \times 100\% \tag{7}$$

$$IoU = \frac{TP}{TP + FP + FN} \times 100\% \tag{8}$$

$$\kappa = \frac{P_0 - P_e}{1 - P_e}$$

$$\text{subject to} \begin{cases} P_e = \dfrac{(TP+FN)(TP+FP)}{\tau^2} + \dfrac{(FN+TN)(FP+TN)}{\tau^2} \\ P_0 = \dfrac{TP+TN}{\tau} \\ \tau = TP+FP+FN+TN \end{cases} \tag{9}$$

In the above equations, *TP* (True Positive) refers to pixels that are accurately segmented as part of the river category; *FP* (False Positive) refers to pixels that are incorrectly classified as part of the river category when they actually belong to a different category; *TN* (True Negative) refers to pixels that are accurately assigned as background pixels (non-river); and *FN* (False Negative) refers to pixels that are incorrectly classified as non-river when they actually belong to the river category.

To interpret the performance of the models quantitatively, we introduced a categorization scheme using predefined thresholds for IoU. This scheme assigns a qualitative label to the model's performance based on observed metric values for each test sample. Accordingly, the model's performance is labeled as "Excellent" when IoU $>$ 0.90, "Good" if $0.85 \leq$ IoU ≤ 0.90, "Fair" if $0.75 \leq$ IoU ≤ 0.85, "Poor" if $0.65 \leq$ IoU < 0.75, and "Unacceptable" otherwise. This approach provides a concise and insightful method of interpreting model performance across key metrics, facilitating a comparative analysis to identify the strengths of DL models.

III. EXPERIMENTS

A. EXPERIMENTAL SETUP

We implemented river water segmentation models in Python, utilizing PyTorch's core library within Google Colab. GPU-accelerated computations in Colab were performed with the following resources: an NVIDIA Tesla T4 (15GB) GPU, an Intel Xeon CPU with two cores running at 2.30 GHz, and 32 GB of RAM. We employed the `segmentation_models_pytorch` library[1] to implement all DL models, excluding SAM, which was fine-tuned based on its original code.[2] All DL models were trained on the datasets without augmentation, using default parameters

[1]https://github.com/qubvel/segmentation_models.pytorch
[2]https://github.com/facebookresearch/segment-anything

Algorithm 1 DL Model Training for River Water Segmentation

Data: $\mathcal{D}_{\text{train}} = \{(I_i, L_i)\}_{i=0}^{N_{\text{train}}-1}$, $\mathcal{D}_{\text{val}} = \{(I_i, L_i)\}_{i=0}^{N_{\text{val}}-1}$, Learning Rate α, Batch size B, Number of epochs N_{epochs}
Result: model f_M, parameters θ_M
Initialization: Initialize model parameters θ_M
Initialize the best validation loss $\mathcal{L}_{\text{val}_{\text{Best}}} \leftarrow \infty$;
Step 1: Data preprocessing: foreach *sample* $(I_i, L_i) \in \mathcal{D}_{train}$ **do**
  $(I_i^P, L_i^P) \leftarrow \text{Preprocess}(I_i, L_i)$; /* by Eq (1) */
  Add (I_i^P, L_i^P) to $\mathcal{D}_{\text{train}}^P$
Preprocess all validation samples: foreach *sample* $(I_i, L_i) \in \mathcal{D}_{val}$ **do**
  $(I_i^P, L_i^P) \leftarrow \text{Preprocess}(I_i, L_i)$; /* by Eq (1) */
  Add (I_i^P, L_i^P) to $\mathcal{D}_{\text{val}}^P$
Step 2: Model Training: for $epoch \leftarrow 1$ **to** N_{epochs} **do**
  foreach *mini-batch* $(I_{i:i+B-1}^P, L_{i:i+B-1}^P) \in \mathcal{D}_{train}^P$ **do**
    $\hat{L}_{i:i+B-1}^P \leftarrow f_M(I_{i:i+B-1}^P; \theta_M)$ $\mathcal{L}_{\text{BCE}_{i:i+B-1}} \leftarrow \text{BCEWithLogitsLoss}(\hat{L}_{i:i+B-1}^P, L_{i:i+B-1}^P)$; /* by Eq (2) */
    $\theta_M \leftarrow \text{AdamOptimizer}(\theta_M, \nabla_{\theta_M} \frac{1}{B} \sum_{j=i}^{i+B-1} \mathcal{L}_{\text{BCE}_j}, \alpha)$
  Calculate Training Loss for Epoch:
  $\mathcal{L}_{\text{train}} \leftarrow \frac{1}{N_{\text{train}}/B} \sum_{i=1}^{N_{\text{train}}/B} \frac{1}{B} \sum_{j=i}^{i+B-1} \mathcal{L}_{\text{BCE}_j}$
  Print($\mathcal{L}_{\text{train}}$);
  Model Evaluation on Validation Set: foreach *mini-batch* $(I_{i:i+B-1}^P, L_{i:i+B-1}^P) \in \mathcal{D}_{val}^P$ **do**
    $\hat{L}_{i:i+B-1}^P \leftarrow f_M(I_{i:i+B-1}^P; \theta_M)$ $\mathcal{L}_{\text{BCE}_{i:i+B-1}} \leftarrow \text{BCEWithLogitsLoss}(\hat{L}_{i:i+B-1}^P, L_{i:i+B-1}^P)$; /* by Eq (2) */
  Calculate Validation Loss for Epoch:
  $\mathcal{L}_{\text{val}} \leftarrow \frac{1}{N_{\text{val}}/B} \sum_{i=1}^{N_{\text{val}}/B} \frac{1}{B} \sum_{j=i}^{i+B-1} \mathcal{L}_{\text{BCE}_j}$
  Print($\mathcal{L}_{\text{val}}$)
  Save the model checkpoint if the validation loss improves if $\mathcal{L}_{val} < \mathcal{L}_{val_{Best}}$ **then**
    $\mathcal{L}_{\text{val}_{\text{Best}}} \leftarrow \mathcal{L}_{\text{val}}$;
    return model f_M, parameters θ_M

Algorithm 2 DL Model Testing for River Water Segmentation

Inputs: $\mathcal{D}_{\text{test}} = \{(I_i, L_i)\}_{i=0}^{N_{\text{test}}-1}$, Trained model f_M, Model parameters θ_M;
Outputs: Predictions on the test dataset $\hat{L}_{\text{test}}$, Average evaluation metrics AverageMetrics;
Preprocess all test samples: foreach *sample* $(I_i, L_i) \in \mathcal{D}_{test}$ **do**
  $(I_i^P, L_i^P) \leftarrow \text{Preprocess}(I_i, L_i)$; /* by Eq (1) */
  Add (I_i^P, L_i^P) to $\mathcal{D}_{\text{test}}^P$
Make Predictions and Evaluate Metrics:
MetricsList $\leftarrow$ [] ; /* List to store individual metrics for each sample */
for $i \leftarrow 0$ **to** $N_{test} - 1$ **do**
  $\hat{L}_i^P \leftarrow f_M(I_i^P; \theta_M)$; /* by the trained model f_M */
  Add $(\hat{L}_i^P)$ to $\hat{L}_{\text{test}}$; /* Add $(\hat{L}_i^P)$ to $\hat{L}_{\text{test}}$ */
  MetricsList.append(EvaluateMetrics($\hat{L}_i^P, L_i^P$)) ; /* by evaluation metrics, reported in Subsection II-D */
Calculate Average Metrics:
AverageMetrics $\leftarrow$ Average(MetricsList) ; /* Calculate the average of the evaluated metrics */
return Predictions on the test dataset $\hat{L}_{\text{test}}$, Average evaluation metrics AverageMetrics;

for 50 epochs with a batch size of 32 and a learning rate of 1×10^{-5}. The model parameters from the epoch with the lowest validation loss were saved as the best-performing ones.

The computational time per epoch and the number of trainable parameters (n_{param}) for each model have been reported in Table 2.

TABLE 1. Amount of training, validation, and test samples of datasets used in this study.

Dataset	Training	Validation	Test
Kaggle WaterNet	523	122	103
Elbersdorf/Wesenitz	500	200	301
RIWA.*v*1	789	228	111
LuFI-RiverSnap.*v*1	657	202	233

As evident from Table 2, the SAM algorithm featured a parameter count roughly three times larger ($>$ 60M) than that of other models, thereby justifying its considerable computational expense during dataset training. PSPNet$_{(\text{ResNet50})}$ and PAN$_{(\text{ResNet50})}$ were among the lighter and more efficient networks, boasting the lowest parameters and computational costs. DeeplabV3+$_{(\text{ResNet50})}$, U-Net$_{(\text{ResNet50})}$, and LinkNet$_{(\text{ResNet50})}$ exhibited roughly the same training time, although DeeplabV3+$_{(\text{ResNet50})}$ was lighter than both. To clarify why we fine-tuned SAM with a ViT (91M) backbone, a comparative analysis was conducted in Appendix A.

TABLE 2. The number of parameters and average training time per epoch for each DL configuration.

Model	n_{param}	Average Training Time per Epoch (sec.)			
		Kaggle WaterNet	Elbersdorf/Wesenitz	RIWA.v1	LuFI-RiverSnap.v1
U-Net$_{(\text{ResNet50})}$	32M	126	184	281	313
PSPNet$_{(\text{ResNet50})}$	25M	97	160	246	289
DeepLabV3+$_{(\text{ResNet50})}$	26M	121	185	297	308
PAN$_{(\text{ResNet50})}$	24M	107	172	263	294
LinkNet$_{(\text{ResNet50})}$	31M	131	190	304	311
SAM	94M	270	319	437	424

TABLE 3. Comparison results of DL segmentation models on the Kaggle WaterNet dataset. (Red: the best, blue: second best).

Model	OA	κ	IoU	$Precision$	$Recall$	F_s
U-Net$_{(\text{ResNet50})}$	0.955	0.889	0.874	0.926	0.919	0.920
PSPNet$_{(\text{ResNet50})}$	0.944	0.867	0.849	0.904	0.916	0.905
DeeplabV3+$_{(\text{ResNet50})}$	0.957	0.900	0.886	0.921	0.941	0.929
PAN$_{(\text{ResNet50})}$	0.953	0.889	0.875	0.921	0.927	0.921
LinkNet$_{(\text{ResNet50})}$	0.956	0.896	0.881	0.919	0.938	0.926
SAM	0.963	0.914	0.901	0.953	0.932	0.939

It's important to highlight that, due to limitations in computational resources, the 512 × 512 input dimensions were used for all model training processes. Its worth noting that the SAM model resamples 512 × 512 images to 1024 × 1024, as its ViT backbone requires this size of inputs. We evaluated the impact of this resampling process on SAM's performance in Appendix B.

We plotted training and validation accuracy graphs to address overfitting and determine optimal training epochs for each DL network, as depicted in Fig. 11.

B. EXPERIMENTAL RESULTS

The following subsections have presented and discussed the visualization and quantitative results of the DL models for each considered dataset.

1) RESULTS OBTAINED ON THE KAGGLE WaterNet DATASET

Fig. 12 shows examples of segmentation results obtained by applying the considered DL models to the Kaggle WaterNet test images. SAM outperformed other models in river water segmentation (see Fig. 12(g)), while PSPNet(ResNet50) showed inferior performance in cases 1, 5, and 9 (see Fig. 12(c)). DeepLabV3+$_{(\text{ResNet50})}$, PAN$_{(\text{ResNet50})}$, and LinkNet$_{(\text{ResNet50})}$ consistently produced robust segmentation results in most instances, except for case 12 (refer to column 12 of Fig. 12 (d), (e), and (f)). In these cases, these models incorrectly identified areas around the river as water. However, PAN$_{(\text{ResNet50})}$ demonstrated better accuracy than DeepLabV3+$_{(\text{ResNet50})}$ and LinkNet$_{(\text{ResNet50})}$ in these cases. The U-Net$_{(\text{ResNet50})}$ model displayed moderate segmentation performance compared to other models, where It exhibited good performance in some cases, like test images 6,7,8, and 11 (see Fig. 12 (b)), and poorer performance in others.

The quantitative results reported in Table 3 closely align with the findings discussed from Fig. 12. The SAM model emerged as the top-performing model among the tested DL models as evidenced by the highest metric values. Following closely behind SAM, DeepLabV3+$_{(\text{ResNet50})}$ exhibited the second-best performance, with only a 1.5% degradation in κ and IoU. LinkNet$_{(\text{ResNet50})}$ also demonstrated robust results, with almost identical results to DeepLabV3+$_{(\text{ResNet50})}$. On the other hand, PSPNet$_{(\text{ResNet50})}$ yielded poorer results, displaying the lowest metric values among the models assessed. Moreover, U-Net$_{(\text{ResNet50})}$ and PAN$_{(\text{ResNet50})}$ showed similar performance in river water segmentation.

The box plots in Fig. 13(a) also confirm the findings. SAM consistently showed higher median values than other models, indicating more consistent performance with fewer fluctuations. SAM's performance distribution was also more concentrated around the median of the metrics, suggesting less susceptibility to outliers and more predictable results. Conversely, PSPNet$_{(\text{ResNet50})}$ had the lowest median values, indicating comparatively less stable performance.

As depicted in Fig. 13(b) SAM achieved excellent performance in approximately 77% of the testing data, with a failure rate of around 6%. Conversely, DeepLabV3+, PAN, LinkNet, and U-Net achieved excellent performance in roughly 69% of the test data, with varying rates of "Fair" and "Good" performance levels. Notably, while both LinkNet and PAN achieved the same number of "Excellent" ratings in water segmentation, PAN had a higher failure rate. Moreover, Fig. 13(b) also confirm PSPNet's inferior performance, where it achieved a success rate of 56%, which was 21% lower than SAM and 10% lower than the other models in river segmentation.

2) RESULTS OBTAINED ON THE ELBERSDORF/WESENITZ DATASET

It is evident from Fig. 14 that all the models qualitatively produced almost identical segmentation results, effectively distinguishing water from the background in the Elbersdorf/Wesenitz dataset. Notably, the considered state-of-the-art models exhibited slightly greater robustness than SAM. SAM's results included limited FN and FP errors attributed to reflected shadows and floating vegetation in the shallow river water. These findings are reinforced by the comparative results of the DL models, as presented in Table 4.

Table 4 shows that the metrics for all models are in the range of 0.99, signifying their robust capability in detecting river water within the Elbersdorf/Wesenitz dataset. This high level of performance is further verified by Fig. 13(b), illustrating that all methods achieved an "Excellent" rating over all 301 test data of this dataset. Upon closer inspection,

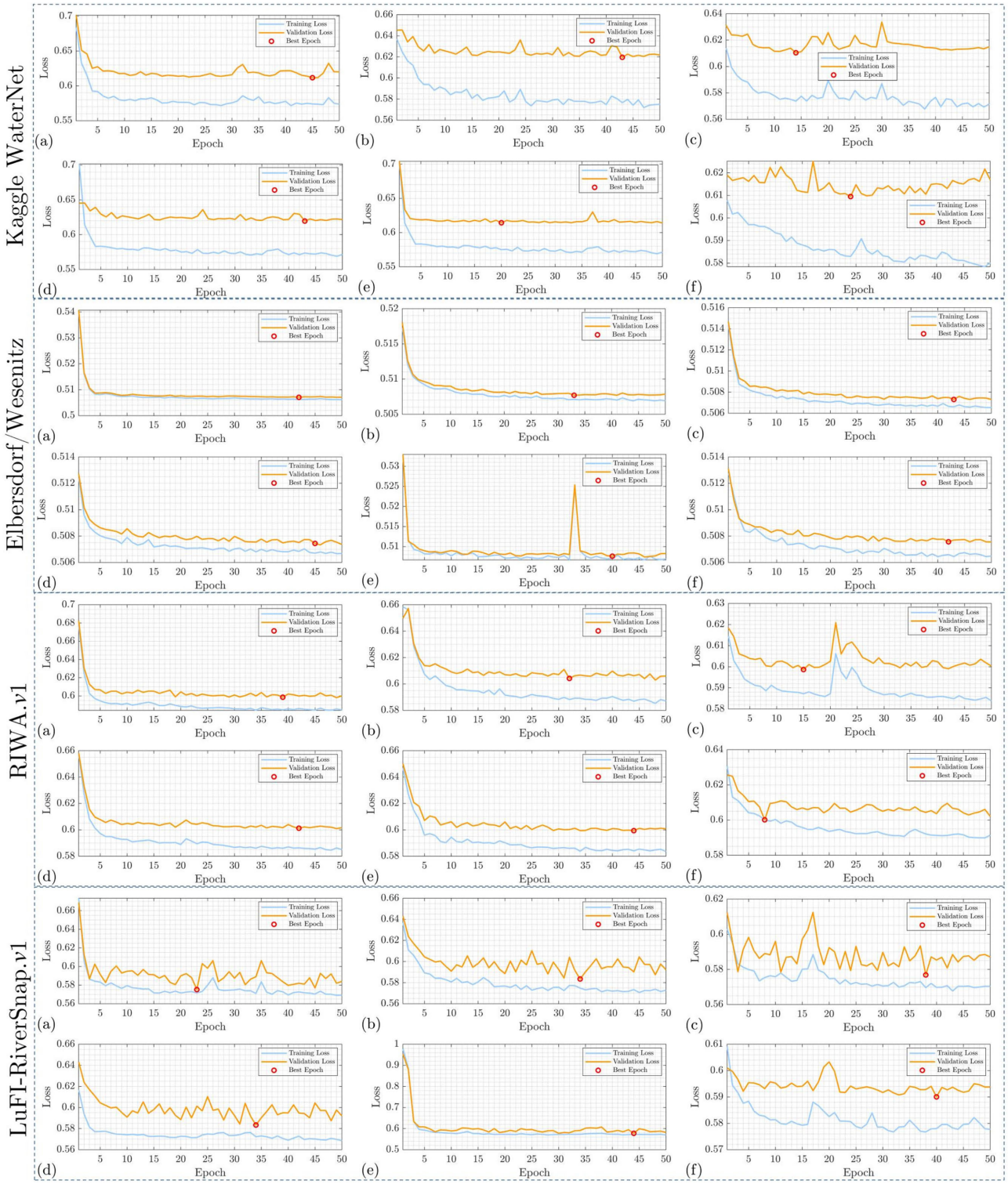


FIGURE 11. Training and validation loss on the considered datasets when (a) U-Net$_{(ResNet50)}$, (b) PSPNet$_{(ResNet50)}$, (c) DeeplabV3+$_{(ResNet50)}$, (d) PAN$_{(ResNet50)}$, (e) LinkNet$_{(ResNet50)}$, and (f) SAM were used as DL models for river water segmentation.

it becomes apparent that SAM's model exhibited a slightly weaker performance compared to other models, as previously noted in the visual analysis of Fig. 14. The box plots in Figure Fig. 13(a) provide clearer insights into differences in model performance. As depicted, U-Net$_{(ResNet50)}$ exhibited superior performance, with the results densely clustered around its median. In contrast, SAM displayed a lower median than the other models, signifying its comparatively weaker performance in river segmentation within this case.

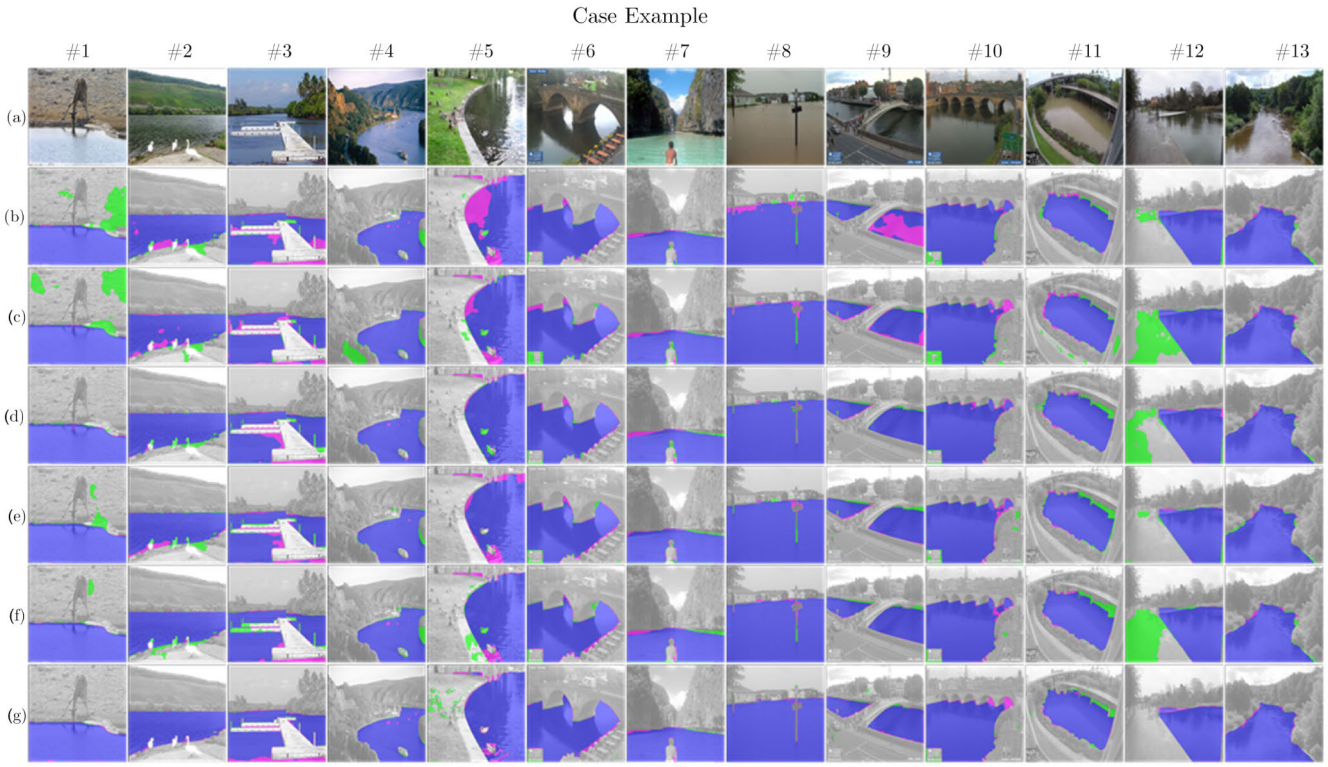


FIGURE 12. Some examples of river water segmentation results on the Kaggle WaterNet dataset. (a) Images and segmentation results generated by (b) U-Net$_{(ResNet50)}$, (c) PSPNet$_{(ResNet50)}$, (d) DeeplabV3+$_{(ResNet50)}$, (e) PAN$_{(ResNet50)}$, (f) LinkNet$_{(ResNet50)}$, and (g) SAM were used as DL models for river water segmentation.Green: False Positives (*FP*) detection, Pink: False Negatives (*FN*) detection, Blue: correct detection of river water.

TABLE 4. Comparison results of DL segmentation models on the Elbersdorf/Wesenitz dataset. (Red: the best, blue: second best).

Model	OA	κ	IoU	$Precision$	$Recall$	F_s
U-Net$_{(ResNet50)}$	0.999	0.998	0.998	0.999	0.999	0.999
PSPNet$_{(ResNet50)}$	0.998	0.995	0.995	0.997	0.999	0.998
DeeplabV3+$_{(ResNet50)}$	0.998	0.997	0.997	0.999	0.998	0.998
PAN$_{(ResNet50)}$	0.998	0.996	0.996	0.998	0.998	0.998
LinkNet$_{(ResNet50)}$	0.998	0.997	0.997	0.998	0.999	0.998
SAM	0.994	0.988	0.988	0.996	0.991	0.994

TABLE 5. Comparison results of DL segmentation models on the RIWA.*v*1 dataset. (Red: the best, blue: second best).

Model	OA	κ	IoU	$Precision$	$Recall$	F_s
U-Net$_{(ResNet50)}$	0.969	0.907	0.878	0.927	0.942	0.930
PSPNet$_{(ResNet50)}$	0.958	0.869	0.834	0.899	0.908	0.899
DeeplabV3+$_{(ResNet50)}$	0.966	0.901	0.873	0.929	0.932	0.926
PAN$_{(ResNet50)}$	0.964	0.892	0.859	0.929	0.916	0.918
LinkNet$_{(ResNet50)}$	0.966	0.900	0.869	0.930	0.928	0.924
SAM	0.966	0.891	0.860	0.943	0.897	0.914

3) RESULTS OBTAINED ON THE RIWA.*v*1 DATASET

In contrast to the previous dataset, all considered models displayed varying performances in segmenting river water in the RIWA.*v*1 dataset, as depicted in Fig. 15. In several instances (cases 4, 5, 6, and 12) of the RIWA.*v*1 dataset, SAM demonstrated superior visual segmentation of river water compared to other models. However, common models exhibited both false positive (*FP*) and false negative (*FN*) errors in identifying the river zone in cases 2 and 5 (refer to columns 2 and 5 of Fig. 15(a)-(f)). Interestingly, in certain instances, particularly cases 8-13, many models performed comparably well or even surpassed SAM in segmenting river water (refer to columns 8-13 of Fig. 15(a)-(g)).

Table 5 provides a comprehensive assessment of the considered segmentation models on all test samples of the RIWA.*v*1 dataset. Upon scrutinizing the results in Table 5, it is evident that the U-Net$_{(ResNet50)}$ model demonstrated the best overall performance in terms of all metrics, except for precision, where SAM achieved a better result. DeepLabV3+$_{(ResNet50)}$ ranked as the second-best model in river water segmentation, with a slight difference (approximately 0.5% on average) compared to U-Net$_{(ResNet50)}$, closely followed by LinkNet$_{(ResNet50)}$. Although SAM and PAN$_{(ResNet50)}$ produced similar segmentation results, SAM visually appeared to have superior performance. Conversely, PSPNet$_{(ResNet50)}$ emerged as the least effective model in the segmentation of river water in the RIWA.*v*1 dataset, with its IoU and κ metrics being almost 4.5% lower than those of the best-performing model.

These results are strongly supported by the boxplots presented in Fig. 13(a), where U-Net$_{(ResNet50)}$ exhibited higher median values, shorter boxes, and fewer outliers than other models, indicating its general robustness regarding κ, IoU, and F_s. Moreover, in 64 test samples, the performance of the U-Net$_{(ResNet50)}$ model was classified as "Excellent," confirming the findings of Table 5 and Fig. 15, as illustrated in Fig. 13 (b). In contrast, PSPNet$_{(ResNet50)}$ displayed the worst performance, as evidenced by lower median values, elongated

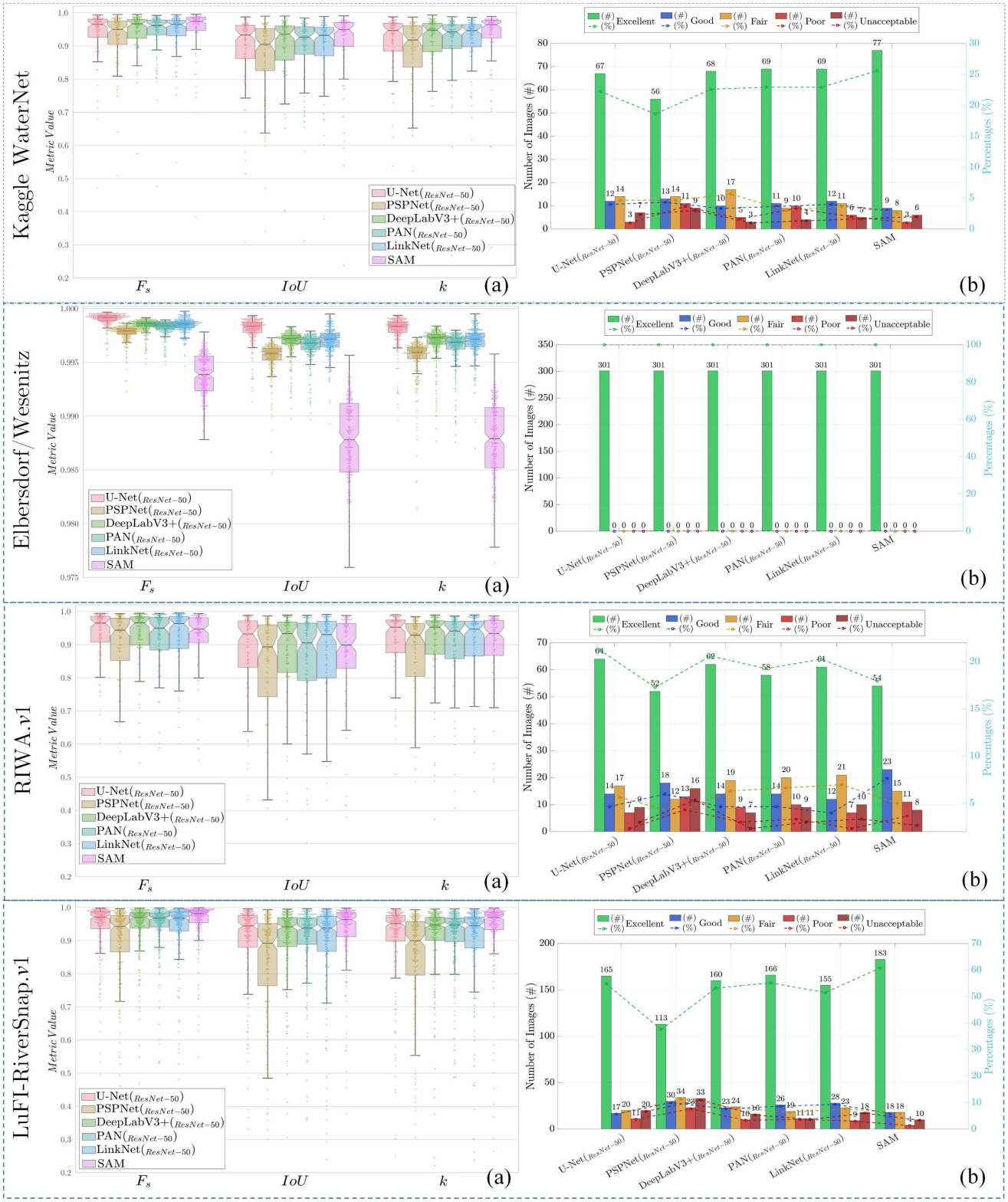


FIGURE 13. (a) Box plots depicting segmentation performance of DL models on the considered datasets, evaluated with κ, *IoU*, and F_S metrics. Each box represents the median, with edges indicating the 25^th and 75^th percentiles, (b) Success levels of DL models on considered datasets using predefined *IoU* thresholds.

boxes, and numerous outliers (see Fig. 13 (a)). Furthermore, it had the lowest rate of successful segmentation compared to others, with its performance classified as ''Excellent'' in less than 50% of test samples (approximately $\sim$ 52 samples) for

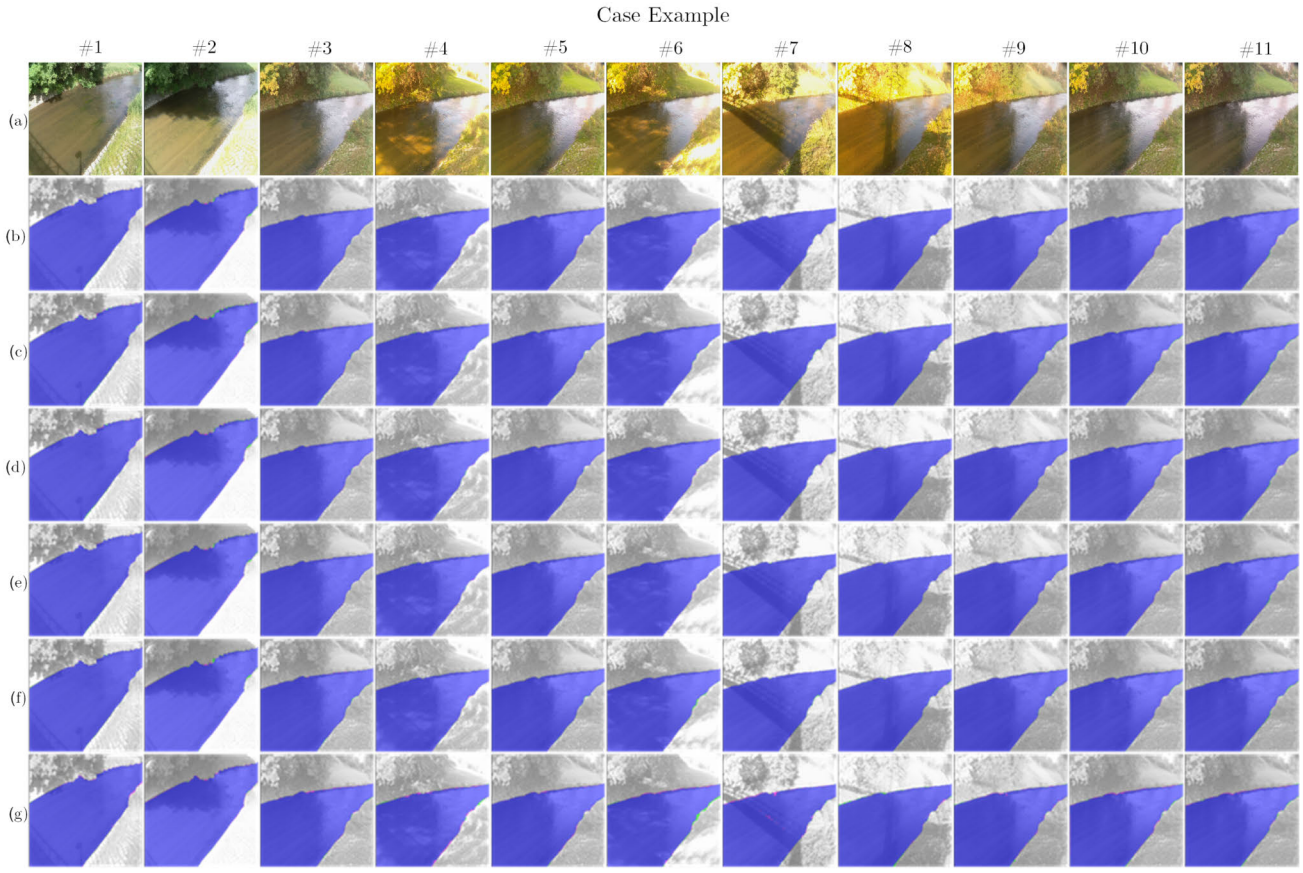


FIGURE 14. Some examples of river water segmentation results on the Elbersdorf/Wesenitz dataset. (a) Images and segmentation results generated by (b) U-Net$_{(ResNet50)}$, (c) PSPNet$_{(ResNet50)}$, (d) DeeplabV3+$_{(ResNet50)}$, (e) PAN$_{(ResNet50)}$, (f) LinkNet$_{(ResNet50)}$, and (g) SAM were used as DL models for river water segmentation.Green: False Positives (*FP*) detection, Pink: False Negatives (*FN*) detection, Blue: correct detection of river water.

the RIWA.*v*1 dataset (see Fig. 13 (b)). Although the SAM model visually appeared to have good performance in the cases presented in Fig. 16, its performance was only classified as "Excellent" in 54 test samples, just 2 samples more than the worst model (PSPNet$_{(ResNet50)}$) and 10 samples lower than the best model (U-Net$_{(ResNet50)}$). However, in 25 test samples, the performance of SAM was classified as "Good," the highest number in this success rate among all considered models.

4) RESULTS OBTAINED ON THE LuFI-RiverSnap.*v*1 DATASET

As expected, the diverse characteristics of the LuFI-RiverSnap.*v*1 dataset led to varying performances among the considered segmentation models, as illustrated in the showcased examples in Fig. 16. In this case, a visual comparison of the segmentation results displayed in Fig. 16 verifies the exceptional capabilities of SAM in providing river water bodies very close to the ground truth. This heightened accuracy is particularly evident in cases 4, 11, and 12, where SAM significantly outperformed other models (refer to columns 4, 11, and 12 in Fig. 16(a)-(g)). The segmentation results obtained from U-Net$_{(ResNet50)}$, LinkNet$_{(ResNet50)}$, DeepLabV3+$_{(ResNet50)}$, and PAN$_{(ResNet50)}$ were closely matched, with PAN$_{(ResNet50)}$ and DeepLabV3+$_{(ResNet50)}$ exhibiting slightly superior visual performance. The distinctions among these models become more apparent in cases 1, 3, and 5, where U-Net$_{(ResNet50)}$, PAN$_{(ResNet50)}$, and LinkNet$_{(ResNet50)}$ demonstrated comparable segmentation results, respectively (see columns 1, 3, and 5 in Fig. 16(a)-(f)). Conversely, in line with previous datasets, PSPNet$_{(ResNet50)}$ exhibited poor performance, consistently failing to accurately detect the river water area in most instances (see Fig. 16(a) and (c)). Table 6 presents the quantitative segmentation results of the models considered for the RiverSnap dataset to enhance the analysis further. As depicted in Table 6, the SAM model demonstrated superior performance to other models in most metrics, except for *OA* and *Recall*, where PAN$_{(ResNet50)}$ and U-Net$_{(ResNet50)}$ achieved slightly better results. DeepLabV3+$_{(ResNet50)}$ was the second-best model for river water segmentation, closely followed by PAN$_{(ResNet50)}$ and U-Net$_{(ResNet50)}$. Notably, LinkNet$_{(ResNet50)}$ also delivered a commendable performance, with metric values slightly close to those of the PAN$_{(ResNet50)}$ and U-Net$_{(ResNet50)}$ methods. On the other hand, PSPNet$_{(ResNet50)}$ had the worst quantitative results regarding all metrics. These quantitative findings align seamlessly with our visual analyses, as highlighted in Fig. 16.

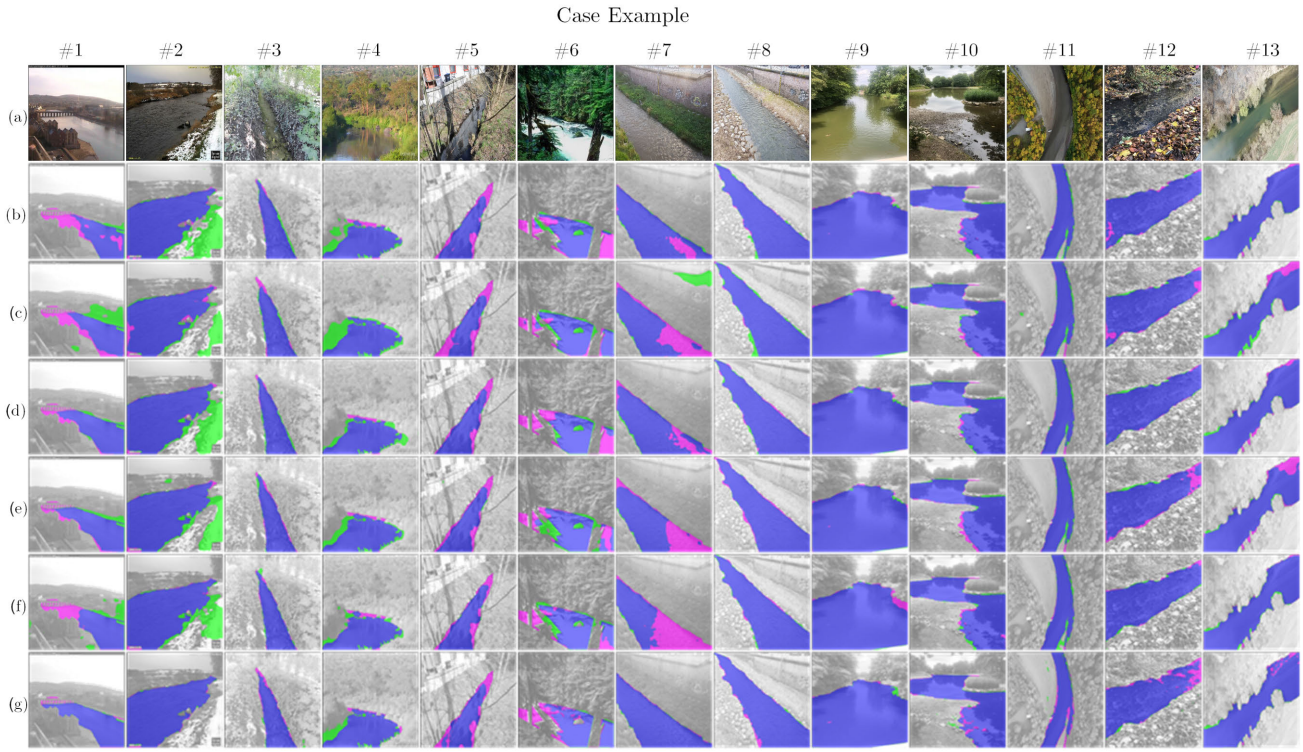


FIGURE 15. Some examples of river water segmentation results on the RIWA.*v*1. (a) Images and segmentation results generated by (b) U-Net$_{(ResNet50)}$, (c) PSPNet$_{(ResNet50)}$, (d) DeeplabV3+$_{(ResNet50)}$, (e) PAN$_{(ResNet50)}$, (f) LinkNet$_{(ResNet50)}$, and (g) SAM were used as DL models for river water segmentation.Green: False Positives (*FP*) detection, Pink: False Negatives (*FN*) detection, Blue: correct detection of river water.

TABLE 6. Comparison results of DL segmentation models on the LuFI-RiverSnap.*v*1 dataset. (Red: the best, blue: second best).

Model	OA	κ	IoU	$Precision$	$Recall$	F_s
U-Net$_{(ResNet50)}$	0.961	0.900	0.890	0.939	0.942	0.934
PSPNet$_{(ResNet50)}$	0.941	0.845	0.833	0.951	0.865	0.899
DeeplabV3+$_{(ResNet50)}$	0.961	0.909	0.898	0.951	0.938	0.941
PAN$_{(ResNet50)}$	0.966	0.904	0.892	0.952	0.938	0.938
LinkNet$_{(ResNet50)}$	0.958	0.891	0.884	0.939	0.932	0.930
SAM	0.963	0.931	0.925	0.985	0.938	0.957

These findings are reinforced by the boxplots presented in Fig. 13(a) and the success levels depicted in Fig. 13(b). For instance, as shown in Fig. 13(a), SAM displayed outstanding performance with higher median values, shorter boxes, and fewer outliers than other models. This remarkable result was corroborated by SAM's results in Fig. 13(b), where it achieved an 'Excellent' rating for approximately 78% of the testing data ($\sim$ 183 test images), with a minimal failure rate of roughly 5%. Following SAM, PAN$_{(ResNet50)}$, and U-Net$_{(ResNet50)}$ showcased commendable performance, both earning an 'Excellent' rating for 66% of the RiverSnap test data. Notably, PAN$_{(ResNet50)}$ exhibited greater robustness compared to U-Net$_{(ResNet50)}$, evidenced by its failure rate being only half that of U-Net$_{(ResNet50)}$, along with higher median values, shorter boxes, and fewer outliers (see Fig. 13(a) and (b)). In contrast, PSPNet$_{(ResNet50)}$'s performance was described as 'Excellent' for only 113 (less than 50%) RiverSnap test samples, as indicated in Fig. 13(b). It also achieved lower median values, larger boxes, and fewer outliers compared to other models, explaining its comparatively poor performance in this case.

C. EXPERIMENTAL ANALYSIS AND DISCUSSION

Using the evaluation framework from the previous subsection, we thoroughly evaluated six DL models across four datasets for river water segmentation. This assessment aimed to uncover performance trends, highlighting each model's strengths and weaknesses across different metrics. This section summarizes the key findings of these evaluations.

In our assessment, SAM demonstrated visually and quantitatively superior water segmentation results on the Kaggle WaterNet and LuFI-RiverSnap.*v*1 datasets compared to other models (see Table 3 and Table 6). This exceptional performance can be attributed to its robust ViT encoder [43], significantly enhancing its capabilities. However, SAM demonstrated comparable performance but did not perform as effectively as most of the considered models in some cases of the RIWA.*v*1 dataset. This can be attributed to the limitations of SAM, as discussed in [44], primarily related to incorrect predictions, broken masks, or significant errors in challenging river scene cases. Additionally, while SAM showed comparable performance in segmenting Elbersdorf/Wesenitz data, it did not perform as well as other models in segmenting rivers with clear water and distinct structures. This stems from SAM's original design, which was developed to accurately segment everything, not specific object segmentation [35].

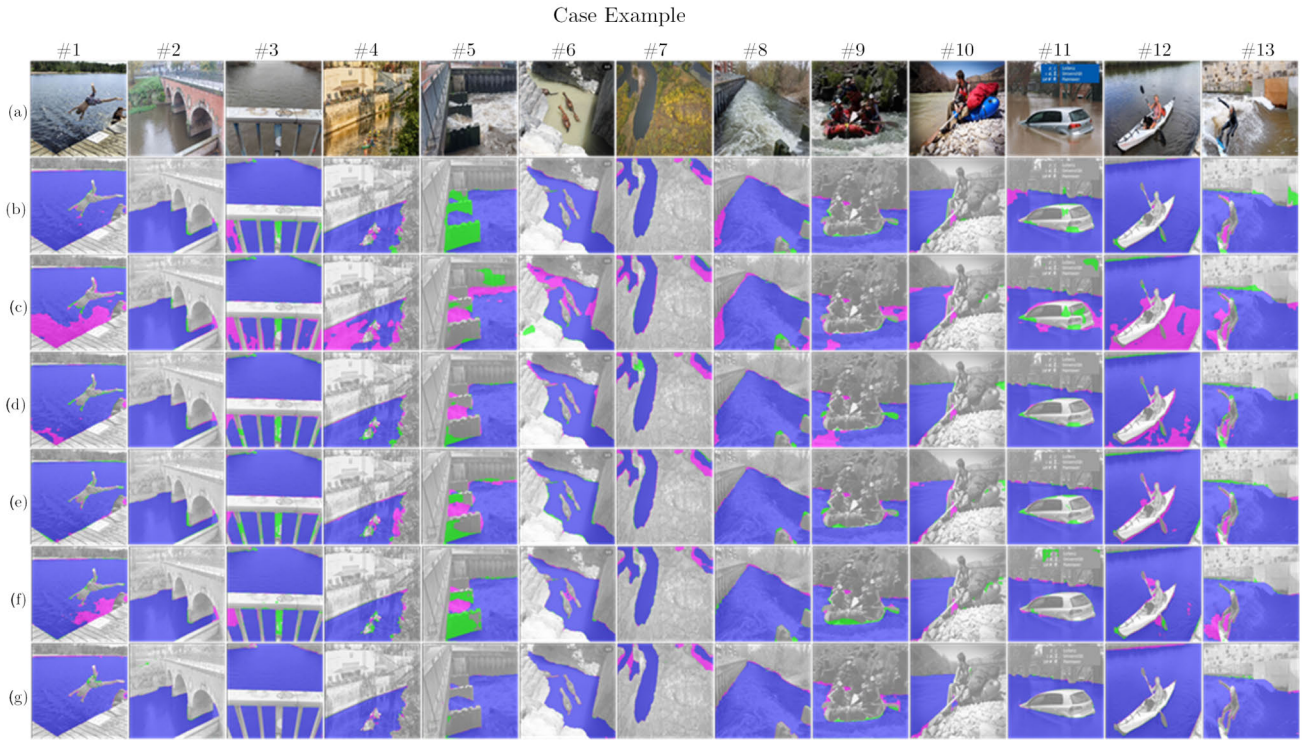


FIGURE 16. Some examples of river water segmentation results on the LuFI-RiverSnap.*v*1. (a) Images and segmentation results generated by (b) U-Net$_{(ResNet50)}$, (c) PSPNet$_{(ResNet50)}$, (d) DeeplabV3+$_{(ResNet50)}$, (e) PAN$_{(ResNet50)}$, (f) LinkNet$_{(ResNet50)}$, and (g) SAM were used as DL models for river water segmentation.Green: False Positives (*FP*) detection, Pink: False Negatives (*FN*) detection, Blue: correct detection of river water.

TABLE 7. Qualitative assessment and summary of the performance of considered DL models for river water segmentation in terms of accuracy for the Entire image (overall) and each test image (image-wise), visual quality, simplicity and speed(computing time), generality.

Model	Quantitative accuracy		Visual performance	Simplicity & Speed	Robustness & Generalization
	Overall	Image-wise			
U-Net$_{(ResNet50)}$	●●●●	●●●	●●●	●●	●●●
PSPNet$_{(ResNet50)}$	●●	●●	●	●●●●	●●
DeeplabV3+$_{(ResNet50)}$	●●●	●●●	●●●	●●●	●●
PAN$_{(ResNet50)}$	●●●	●●●●	●●●	●●●●	●●
LinkNet$_{(ResNet50)}$	●●●	●●●	●●●	●●	●●
SAM	●●●●	●●●●	●●●●	●	●●●●

One bullet denotes the worst performance, while four bullets refers to the best performance.

The U-Net$_{(ResNet50)}$ was the best model in river water segmentation for the RIWA.*v*1, which aligns with the finding in [3] in which this model was the best on this dataset. Furthermore, all models demonstrated promising performance in the Elbersdorf/Wesenitz dataset. This can be attributed to the fact that this data includes images captured by a sensor with specific characteristics in a distinct river scene over time, resulting in limited variability. This characteristic makes these data well-suited for adaptation by most deep learning methods, as observed in the comparable results achieved by SegNet and FCN models presented in [11]. The PSPNet$_{(ResNet50)}$ yielded the poorest results in river water segmentation in all cases, primarily due to its comparatively lightweight architecture compared to the other models tested in this study.

To synthesize the findings presented in the previous section and the observations outlined above, a comprehensive summary of the performance analysis of the evaluated DL models has been presented in Table 7. SAM performed exceptionally well in all aspects but ranked the lowest among the DL models in terms of simplicity and computation time. Hence, it proves to be a suitable choice for river segmentation tasks in which both accuracy and stability are of interest. U-Net$_{(ResNet50)}$ exhibited a well-balanced performance (moderate to high scores across all criteria) across various criteria. This makes it an adaptable choice for general river segmentation tasks. PAN$_{(ResNet50)}$, DeepLabV3+$_{(ResNet50)}$, and LinkNet$_{(ResNet50)}$ provided a reasonable balance between qualitative/quantitative accuracy and speed. This makes them well suited for segmenting river waters, where a

trade-off between these factors is acceptable. Although the PSPNet$_{(ResNet50)}$ model was the fastest and the easiest to use, it was inferior in other categories, making it less suitable for more demanding applications of water segmentation.

The prospects of this work revolve around the application of DL segmentation models not only for river water body detection but also for a broader range of tasks, including general water body detection, wetland or lake scene assessment, urban flood extent assessment, as well as water level detection from close-range RS images. Although the comparative analysis we presented yielded promising results, it is important to acknowledge certain limitations. For instance, the decision to resize images to 512 × 512 pixels was driven by our resource constraints, enabling us to conduct DL model comparisons. However, future research can delve into the impact of image size on model performance and explore alternative approaches, especially under different resource scenarios. One such avenue could involve splitting inputs into overlapped/no overlapped patches, as discussed in [3], but our computational resources constrained us from pursuing this path. Moreover, comparison with multi-scale fusion backbones-based methods [45], [46], which are known to excel in segmenting large instances, can provide further insights into the performance of DL segmentation models for water segmentation.

Additionally, data augmentation could be a valuable tool in tasks involving river water segmentation under diverse environmental conditions, lighting variations, or other challenges, as demonstrated in [3]. However, our focus on the intrinsic capabilities of DL models for river water segmentation, coupled with limitations in computation units, led us to forego data augmentation in this comparison. While confident that identified limitations have not hindered the study's primary objectives, future research could benefit from additional controls.

IV. CONCLUSION

This study presented an exhaustive comparison of U-Net, PSPNet, DeeplabV3+, PAN, and LinkNet backboned with predefined ResNet50 and SAM in the river water segmentation task. The experimental results were conducted on three benchmark river water segmentation and a newly introduced LuFI-RiverSnap.*v*1 dataset. These experiments provide rich information on the efficacy and adaptability of the models and valuable insights for future advancements in river water segmentation research and applications. The experimental results show that considered DL models can satisfactorily result in river water segmentation from close-range RS images with high variations in water color, illumination, sky, and structural reflection on the surface water. The SAM and U-Net$_{(ResNet50)}$ were more accurate on average than the other tested models in river water segmentation; however, both were slower in computing. Pan, DeepLabV3+$_{(ResNet50)}$, and LinkNet$_{(ResNet50)}$ also achieved a good tradeoff between accuracy and computing time. Moreover, PSPNet$_{(ResNet50)}$ was the worst model among the tested models, although it was the most efficient model regarding computation time.

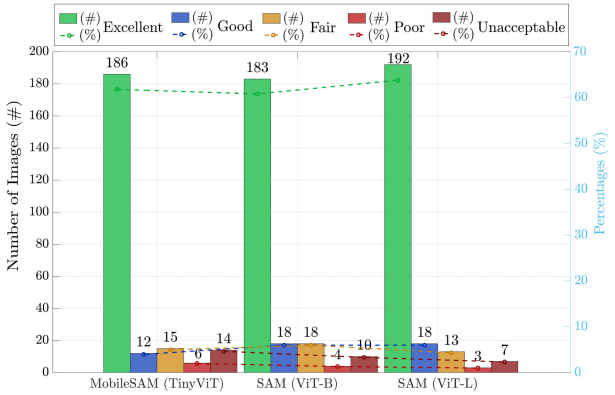


FIGURE 17. Success levels of fine-tuned MobileSAM (TinyViT), SAM (ViT-B), and SAM (ViT-L) models on the LuFI-RiverSnap.*v*1 dataset using predefined *IoU* thresholds.

The LuFI-RiverSnap.*v*1 dataset provided in this study can fulfill the requirement for river water segmentation from close-range RS images, rivers with various colors, and structural reflections. This dataset can support analyses such as trend analysis in river water levels, short-term monitoring of river overflows, and flood extent detection from close-range images. While the dataset is currently undergoing refinement, future versions aim to include a more diverse range of river scene images from around the world. Additionally, the fine-tuning SAM process discussed in this study is not limited to river water segmentation but can be applied or/and integrated with other DL models for diverse object segmentation tasks. Furthermore, As we discard the prompt encoder from the SAM segmentation for a fair comparison, one can explore advanced promptable segmentation DL models to achieve better results in the task of river water segmentation.

APPENDIX A

To validate the effectiveness of employing a ViT (91M) backbone in SAM fine-tuning, we compared its performance with larger ViT (315M) backbone as well as MobileSAM with TinyViT (5.3 M) backbone [47] on LuFI-RiverSnap.*v*1 dataset. The number of parameters and average training time per epoch have been reported in Table 8, while the comparative performance results quantitatively and qualitatively have been presented in Table 9, Fig. 17, and Fig. 18.

As can be seen from Table 9 and Fig. 18, SAM (ViT-L) outperformed SAM (ViT-B) and MobileSAM (TinyViT) in both quantitative and qualitative measures. Specifically, SAM (ViT-L) demonstrated superior performance in 192 test cases, showcasing its capability in accurate river water segmentation (see Fig. 17). However, SAM (ViT-L) required significantly more computational time per epoch (five times longer than SAM (ViT-B) and twenty times longer than SAM (TinyViT)), making it less practical for river water segmentation (see Table 8).

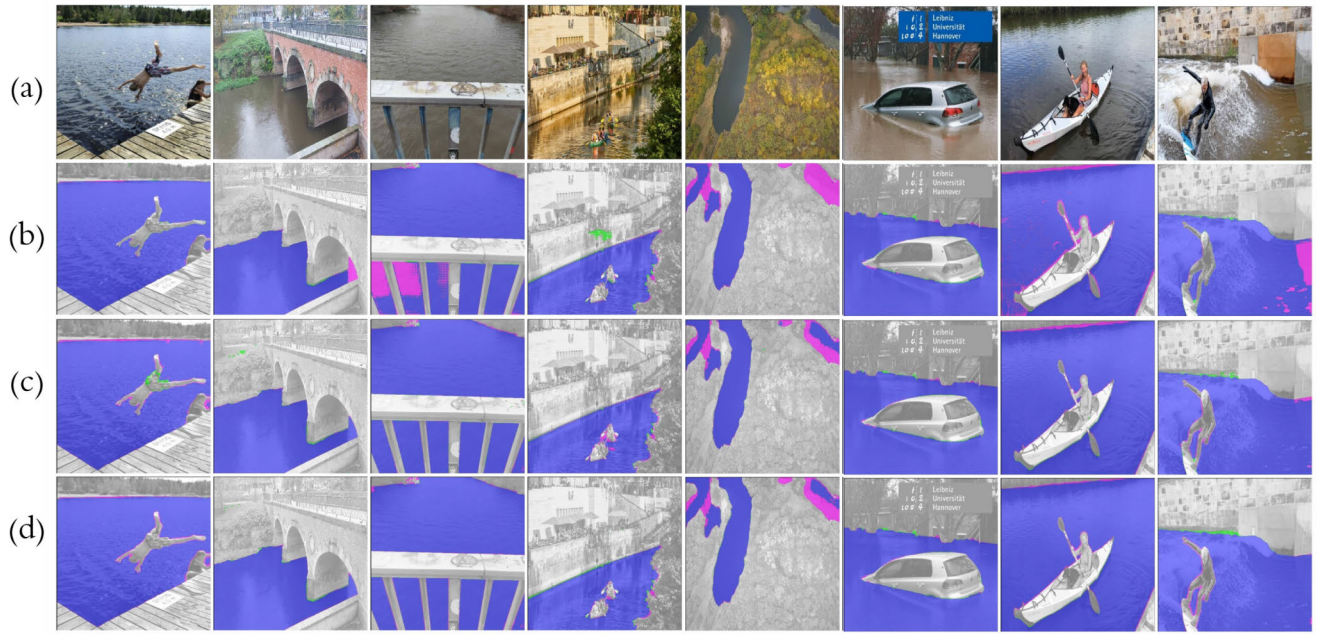


FIGURE 18. Some examples of river water segmentation results on the LuFI-RiverSnap.*v*1. (a) Images and segmentation results generated by (b) MobileSAM (TinyViT), (c) SAM (ViT-B), (d) and SAM (ViT-L).

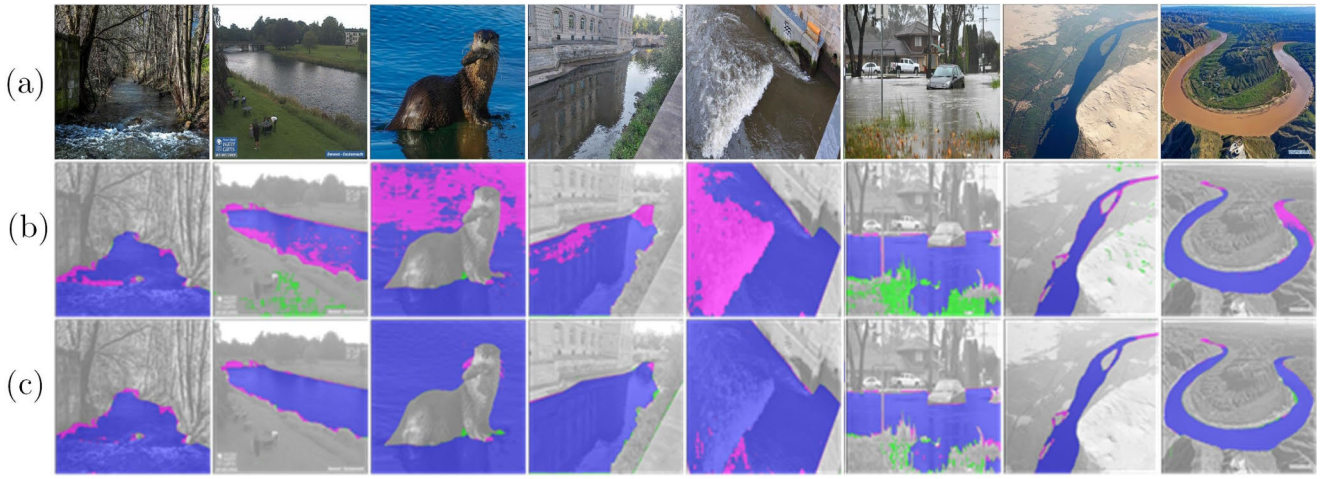


FIGURE 19. (a) Example test images from the LuFI-RiverSnap.*v*1 dataset, predicted by (b) fine-tuned SAM on 512 × 512, and (c) fine-tuned SAM on the input original size.

TABLE 8. The number of parameters and average training time per epoch for MobileSAM (TinyViT), SAM (ViT-B), and SAM (ViT-L).

Model	n_{param}	Average Training Time per Epoch (sec.)
MobileSAM (TinyViT)	10M	45
SAM (ViT-B)	94M	424
SAM (ViT-L)	312M	1185

On the other hand, SAM (TinyViT) emerged as the most time-efficient method for river water segmentation (see Table 8). However, this efficiency came at a cost, as SAM (TinyViT) exhibited weaker performance both quantitatively and qualitatively compared to SAM (ViT-L) and SAM (ViT-B) (see Table 9 and Fig. 18)). Moreover, although SAM (TinyViT) achieved a higher success rate in segmenting river water than SAM (ViT-B), it led to failure results in the same proportion (see Fig. 17).

TABLE 9. Comparison results of fine-tuned MobileSAM (TinyViT), SAM (ViT-B), and SAM (ViT-L) models on the LuFI-RiverSnap.*v*1 dataset. (Red: the best, blue: second best).

Model	OA	κ	IoU	$Precision$	$Recall$	F_s
MobileSAM (TinyViT)	0.960	0.903	0.899	0.965	0.912	0.928
SAM (ViT-B)	0.963	0.931	0.925	0.985	0.938	0.957
SAM (ViT-L)	0.979	0.940	0.931	0.972	0.948	0.958

Conversely, the quantitative and qualitative results of SAM (ViT-B) closely aligned with those of SAM (ViT-L), but were achieved in significantly less time (see Table 8, Table 9 and

Fig. 18). Therefore, a trade-off between accuracy and time became more apparent when fine-tuning SAM with ViT-B, making it the preferred choice in our comparative study.

APPENDIX B

In our workflow, we employed inputs with a size of 512×512, which were subsequently upsampled to 1024 × 1024 to meet SAM's requirements. SAM was also trained using inputs at their original size from the LuFI-RiverSnap.*v*1 dataset to improve the understanding of this issue (see Fig. 19).

The results illustrated in Fig. 19 clearly showcase accurate segmentation of river water, particularly in challenging cases, when SAM was fine-tuned using the original image size rather than the 512 × 512 input images. Importantly, this improvement is not unique to the SAM model; instead, it is expected that the performance of other models tested would similarly benefit from training on the original input size.

REFERENCES

[1] K. E. Limburg, D. P. Swaney, and D. L. Strayer, "River ecosystems," in *Encyclopedia of Biodiversity*. Amsterdam, The Netherlands: Elsevier, 2001, pp. 213–231. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B0122268652002388

[2] Í. Güneralp, A. M. Filippi, and B. U. Hales, "River-flow boundary delineation from digital aerial photography and ancillary images using support vector machines," *GIScience Remote Sens.*, vol. 50, no. 1, pp. 1–25, Feb. 2013.

[3] F. Wagner, A. Eltner, and H.-G. Maas, "River water segmentation in surveillance camera images: A comparative study of offline and online augmentation using 32 CNNs," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 119, May 2023, Art. no. 103305. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1569843223001279

[4] M. Elias, C. Kehl, and D. Schneider, "Photogrammetric water level determination using smartphone technology," *Photogrammetric Rec.*, vol. 34, no. 166, pp. 198–223, Jun. 2019.

[5] M. Elias and H.-G. Maas, "Measuring water levels by handheld smartphones—A contribution to exploit crowdsourcing in the spatio temporal densification of water gauging networks," *Int. Hydrographic Rev.*, vol. 27, pp. 9–22, May 2022.

[6] X. Blanch, F. Wagner, R. Hedel, J. Grundmann, and A. Eltner, "Towards automatic real-time water level estimation using surveillance cameras," in *Proc. EGU Gen. Assem. Conf. Abstr.*, 2022, Paper no. EGU22-3225.

[7] Y. Guo, Y. Liu, T. Georgiou, and M. S. Lew, "A review of semantic segmentation using deep neural networks," *Int. J. Multimedia Inf. Retr.*, vol. 7, no. 2, pp. 87–93, 2018.

[8] I. Ahmed, M. Ahmad, F. A. Khan, and M. Asif, "Comparison of deep-learning-based segmentation models: Using top view person images," *IEEE Access*, vol. 8, pp. 136361–136373, 2020.

[9] J. Yu, Y. Lin, Y. Zhu, W. Xu, D. Hou, P. Huang, and G. Zhang, "Segmentation of river scenes based on water surface reflection mechanism," *Appl. Sci.*, vol. 10, no. 7, p. 2471, Apr. 2020.

[10] A. Rankin and L. Matthies, "Daytime water detection based on color variation," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2010, pp. 215–221.

[11] A. Eltner, P. O. Bressan, W. N. Gonçalves, T. Akiyama, and J. M. Junior, "Using deep learning for automatic water stage measurements," *Water Resour. Res.*, vol. 57, no. 3, Mar. 2021, Art. no. e2020WR027608, doi: 10.1029/2020WR027608.

[12] N. A. Muhadi, A. F. Abdullah, S. K. Bejo, M. R. Mahadi, and A. Mijic, "Deep learning semantic segmentation for water level estimation using surveillance camera," *Appl. Sci.*, vol. 11, no. 20, p. 9691, Oct. 2021.

[13] Y. Chen, R. Fan, X. Yang, J. Wang, and A. Latif, "Extraction of urban water bodies from high-resolution remote-sensing imagery using deep learning," *Water*, vol. 10, no. 5, p. 585, May 2018.

[14] M. Li, P. Wu, B. Wang, H. Park, H. Yang, and Y. Wu, "A deep learning method of water body extraction from high resolution remote sensing images with multisensors," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 3120–3132, 2021.

[15] W. Feng, H. Sui, W. Huang, C. Xu, and K. An, "Water body extraction from very high-resolution remote sensing imagery using deep U-Net and a superpixel-based conditional random field model," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 4, pp. 618–622, Apr. 2019.

[16] L. Lopez-Fuentes, C. Rossi, and H. Skinnemoen, "River segmentation for flood monitoring," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2017, pp. 3746–3749.

[17] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[18] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional DenseNets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1175–1183.

[19] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1125–1134.

[20] J. Pan, Y. Yin, J. Xiong, W. Luo, G. Gui, and H. Sari, "Deep learning-based unmanned surveillance systems for observing water levels," *IEEE Access*, vol. 6, pp. 73561–73571, 2018.

[21] M. M. de Vitry, S. Kramer, J. D. Wegner, and J. P. Leitão, "Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network," *Hydrol. Earth Syst. Sci.*, vol. 23, no. 11, pp. 4621–4634, Nov. 2019.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. 18th Int. Conf. Med. Image Comput. Comput.-Assist. Intervent. (MICCAI)*. Springer, Oct. 2015, pp. 234–241.

[23] T. S. Akiyama, J. M. Junior, W. N. Gonçalves, P. O. Bressan, A. Eltner, F. Binder, and T. Singer, "Deep learning applied to water segmentation," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 43, pp. 1189–1193, Aug. 2020.

[24] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder–decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[25] R. Vandaele, S. L. Dance, and V. Ojha, "Deep learning for automated river-level monitoring through river-camera images: An approach based on water segmentation and transfer learning," *Hydrol. Earth Syst. Sci.*, vol. 25, no. 8, pp. 4435–4453, Aug. 2021.

[26] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun, "Unified perceptual parsing for scene understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 418–434.

[27] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[29] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 801–818.

[30] A. Eltner, P. Zamboni, R. Hedel, J. Grundmann, and X. Blanch, "Image-based methods for real-time water level estimation," in *EGU General Assembly*, Vienna, Austria, Apr. 2023, Paper EGU23-6745, doi: 10.5194/egusphere-egu23-6745.

[31] X. Blanch, F. Wagner, and A. Eltner. (2022). *RIWA Dataset*. [Online]. Available: https://www.kaggle.com/dsv/4289421

[32] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6230–6239.

[33] H. Li, P. Xiong, J. An, and L. Wang, "Pyramid attention network for semantic segmentation," 2018, *arXiv:1805.10180*.

[34] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.

[35] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," 2023, *arXiv:2304.02643*.

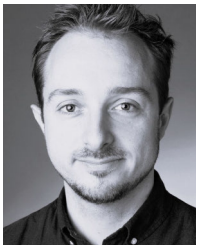
[36] Y. Liang, N. Jafari, X. Luo, Q. Chen, Y. Cao, and X. Li, "WaterNet: An adaptive matching pipeline for segmenting water with volatile appearance," *Comput. Vis. Media*, vol. 6, no. 1, pp. 65–78, Mar. 2020.

[37] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[38] J. Laitala and L. Ruotsalainen, "Computer vision based planogram compliance evaluation," *Appl. Sci.*, vol. 13, no. 18, p. 10145, Sep. 2023.
[39] P. Zhang, Y. Ban, and A. Nascetti, "Learning U-Net without forgetting for near real-time wildfire monitoring by the fusion of SAR and optical time series," *Remote Sens. Environ.*, vol. 261, Aug. 2021, Art. no. 112467.
[40] F. Rajič, L. Ke, Y.-W. Tai, C.-K. Tang, M. Danelljan, and F. Yu, "Segment anything meets point tracking," 2023, *arXiv:2307.01197*.
[41] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[42] Y. Feng and M. Sester, "Extraction of pluvial flood relevant volunteered geographic information (VGI) by deep learning from user generated texts and photos," *ISPRS Int. J. Geo-Inf.*, vol. 7, no. 2, p. 39, Jan. 2018.
[43] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.
[44] L. Ke, M. Ye, M. Danelljan, Y. Liu, Y.-W. Tai, C.-K. Tang, and F. Yu, "Segment anything in high quality," 2023, *arXiv:2306.01567*.
[45] Y. Zhang, J. Chu, L. Leng, and J. Miao, "Mask-refined R-CNN: A network for refining object details in instance segmentation," *Sensors*, vol. 20, no. 4, p. 1010, Feb. 2020.
[46] W. Lin, J. Chu, L. Leng, J. Miao, and L. Wang, "Feature disentanglement in one-stage object detection," *Pattern Recognit.*, vol. 145, Jan. 2024, Art. no. 109878.
[47] C. Zhang, D. Han, Y. Qiao, J. U. Kim, S.-H. Bae, S. Lee, and C. S. Hong, "Faster segment anything: Towards lightweight SAM for mobile applications," 2023, *arXiv:2306.14289*.



ARMIN MOGHIMI received the M.S. and Ph.D. degrees in civil photogrammetry and remote sensing engineering from the K. N. Toosi University of Technology, Tehran, Iran, in 2015 and 2022, respectively.

Since 2023, he has been contributing as a Postdoctoral Research Associate with Ludwig-Franzius-Institute for Hydraulic, Estuarine and Coastal Engineering, Faculty of Civil Engineering and Geodesy, Leibniz University Hannover, specializing in estuary and coastal engineering. His research interests include deep learning, change detection, computer vision, remote sensing techniques, image registration, machine learning, SAR image processing, and LiDAR data processing.



MARIO WELZEL received the Ph.D. degree from Ludwig-Franzius-Institute for Hydraulic, Estuarine and Coastal Engineering, Leibniz University Hannover, Germany, in 2021.

Since then, he has been actively engaged as a Postdoctoral Researcher, a Senior Researcher, and the Project Lead of Ludwig Franzius Institute. His research interests include marine renewable energy, wave-current structure, wave-current structure soil interaction, flood forecasting, and flood management, including numerical and physical models. In addition to his involvement in hydraulic projects, he is currently focused on extracting hydrological parameters from smartphone and drone images using artificial intelligence (AI), to advance classic research areas, such as hydraulic engineering, flood forecast, flood management, and digitalization of the water sector.



TURGAY CELIK received the Ph.D. degree from the University of Warwick, Coventry, U.K., in 2011.

Currently, he is a Professor of digital transformation and the Director of the Wits Institute of Data Science, University of the Witwatersrand Johannesburg, South Africa. His research interests include computer vision, (explainable) artificial intelligence, (health) data science, data-driven optimal control, and remote sensing. He is an Associate Editor of *BMC Medical Informatics and Decision Making*, *IET ELL*, IEEE Access, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing (IEEE JSTARS), and *SIVP* (Springer).



TORSTEN SCHLURMANN received the Ph.D. degree in coastal and maritime engineering from Bergische Universität Wuppertal, in 1999.

He finalized his habilitation thesis to obtain the Venia Legendi in Hydraulic Engineering and Water Resources, in 2005. Following the devastating tsunami in Indian Ocean, in December 2004, he accepted a full-time appointment with the Institute for Environment and Human Security (EHS), Bonn, United Nations University (UNU), Tokyo, and joined the United Nations. He continued his academic career as a Postdoctoral Scientist (Oberingenieur, C2) with Bergische Universität Wuppertal. As the Head of the Section for Coastal Risks, he played a leading role in the design and implementation of a tsunami early warning system (TEWS) in Indian Ocean and was one of the responsible Lead-PI of the two BMBF projects GITEWS (FKZ: 03TSU01) and Last-Mile–Evacuation (FKZ: 03G0666A-E). He soon was appointed as a W3-Professor for hydraulic and coastal engineering with Leibniz University Hannover, in 2007, but remained ex officio formal advisor to the Director of the Institute for Environment and Human Security, Bonn, United Nations University, Tokyo, the United Nations in several overarching UN committees and working groups on the development and operation of TEWS, until 2010. He has been a Professor and the Managing Director of Ludwig-Franzius-Institute for Hydraulic, Estuarine and Coastal Engineering, since 2007. He is currently the Managing Director of the Coastal Research Centre, a joint central institution of Leibniz Universität Hannover and Technische Universität Braunschweig that operates the recently extended, world-renowned large wave-current flume (GWK+). His main research areas concentrate on the modeling of estuarine and coastal processes, strategies, and measures in coastal protection, port construction and maintenance, development of maritime technologies, and risk management. His main research interests include offshore wind and marine renewable energies, the implication of climate-driven processes and impacts from sea level rise in coastal and estuarine environments, the development of nature-based Solutions (NbS), and the quantification of ecosystem services in coastal protection.