JARS Journal of Applied Remote Sensing

RESEARCH PAPER

# Man-made object segmentation around reservoirs by an end-to-end two-phase deep learning-based workflow

**Nayereh Hamidishad*** **and Roberto Marcondes Cesar Jr.**
University of São Paulo, Institute of Mathematics and Statistics, São Paulo, Brazil

**ABSTRACT.** Reservoirs are fundamental infrastructures for the management of water resources. Constructions around them can negatively impact their water quality. Such constructions can be detected by segmenting man-made objects around reservoirs in the remote sensing (RS) images. Deep learning (DL) has attracted considerable attention in recent years as a method for segmenting the RS imagery into different land covers/uses and has achieved remarkable success. We develop an approach based on DL and image processing techniques for man-made object segmentation around the reservoirs. In order to segment man-made objects around the reservoirs in an end-to-end procedure, segmenting reservoirs and identifying the region of interest (RoI) around them are essential. In the proposed two-phase workflow, the reservoir is initially segmented using a DL model, and a postprocessing stage is proposed to remove errors, such as floating vegetation in the generated reservoir map. In the second phase, the RoI around the reservoir (RoIaR) is extracted using the proposed image processing techniques. Finally, the man-made objects in the RoIaR are segmented using a DL model. To illustrate the proposed approach, our task of interest is segmenting man-made objects around some of the most important reservoirs in Brazil. Therefore, we trained the proposed workflow using collected Google Earth images of eight reservoirs in Brazil over two different years. The U-Net-based and SegNet-based architectures are trained to segment the reservoirs. To segment man-made objects in the RoIaR, we trained and evaluated four architectures: U-Net, feature pyramid network, LinkNet, and pyramid scene parsing network. Although the collected data are highly diverse (for example, they belong to different states, seasons, resolutions, etc.), we achieved good performances in both phases. The $F_1$-score of phase-1 and phase-2 highest performance models in segmenting test sets are 96.53% and 90.32%, respectively. Furthermore, applying the proposed postprocessing to the output of reservoir segmentation improves the precision in all studied reservoirs except two cases. We validated the prepared workflow with a reservoir dataset outside the training reservoirs. The $F_1$-scores of the phase-1 segmentation stage, postprocessing stage, and phase-2 segmentation stage are 92.54%, 94.68%, and 88.11%, respectively, which show high generalization ability of the prepared workflow.



**Keywords:** remote sensing imagery; reservoir segmentation; man-made object segmentation; deep learning

Paper 230167G received Apr. 13, 2023; revised Nov. 13, 2023; accepted Dec. 27, 2023; published Jan. 24, 2024.

*Address all correspondence to Nayereh Hamidishad, nhamidi@ime.usp.br

## 1 Introduction

Reservoirs reduce the effects of interseasonal and interannual stream flow fluctuations and hence facilitate water supply, hydroelectric power generation, and flood control, to name a few.[1] There are significant interactions between the environment and reservoirs as essential water resource management tools. For example, the reservoir affects the quality of the water downstream of its dam, and human activities affect the quality of the water in the reservoir as well as the chemical and biological processes in it.[2]

The pixel-based, object-based (OB), and, recently, deep learning (DL) methods are three fundamental approaches that are implemented for mapping remote sensing (RS) images into different land covers and land uses. Pixel-based methods (e.g., support vector machine) rely on the spectral signatures of individual pixels, and each pixel is independently classified.[3] With the increase in the spatial resolution of satellite images resulting from improvements in RS systems, a single pixel does not adequately capture the characteristics of target objects, leading to a reduction in the classification accuracy using pixel-based methods.[4] Over the last decades, the RS community has undertaken considerable efforts to promote the use of OB technology for land cover/use mapping.[5,6] In contrast with pixel-based methods, OB classification methods are less sensitive to the spectral variance within the objects. They can use both object features and spatial relations between the objects. However, the popularity of OB methods is affected by two factors: (1) the majority of them rely on pricey commercial software and (2) the result is highly influenced by parameter selection.

DL has made significant strides in recent years, enabling high-level feature extraction to be carried out automatically while displaying promising results in various domains, including image semantic segmentation. Recently, convolutional neural networks (CNNs) have been among the most advanced algorithms in segmenting RS images into different land covers/uses, and their superior performances compared to traditional methods have been proved.[7,8] The decoder–encoder networks and spatial pyramid pooling-based networks are two state-of-the-art and widely used categories of CNNs. The decoder–encoder-based networks consist of an encoder path and a decoder path. The encoder path consists of convolutional layers to extract the feature maps. Next, these features are transformed/upsampled to dense label maps in the decoder path. Building upon this architecture, networks such as U-Net, SegNet, and feature pyramid network (FPN) have demonstrated strong performances and are frequently utilized in segmenting RS imagery.[9,10] The spatial pyramid pooling-based networks contain a pyramid pooling module to collect multilevel global information from the input image. Pyramid scene parsing network (PSPNet) proposed by Ref. 11 is a broadly adopted architecture in this category.[12]

Semantic segmentation of water bodies using the DL approach is studied in several works. For example, a DL encoder–decoder framework is proposed by Li et al.[13] to extract water bodies from 4-band RS images with resolutions $>1$ m. Chen et al.[14] combined an enhanced super-pixel method with DL to extract urban water bodies from multispectral bands with low spatial resolutions ($>4$ m). The RapidEye 5 m resolution images are used by Ref. 15 to segment gorges reservoir areas to water bodies and other land covers by DL methods. Van Soesbergen et al.[16] proposed a pipeline where a DL model in the first-stage segments the water bodies in moderate spatial resolution RS images. Next, bounding boxes of individual water bodies are classified into two classes, dam reservoir and natural water, by a classifier.

DL is also popular among studies on semantic segmentation of man-made objects in RS images. For example, a DL-based approach is proposed by Ref. 17 to segment ROSIS hyperspectral images as man-made and non-man-made. The man-made class in this work consists of asphalt, metal sheets, bricks, bitumen, and tiles. Before feeding to the network, the data are preprocessed by randomized principal component analysis for input dimension reduction. Residential land, industrial land, traffic land, woodland, and unused land are five defined classes by Ref. 18 for collected RGB images with 0.5 m resolution. To segment images, they proposed a workflow in which the images are fed to two networks in parallel. Next, their output feature maps are fused to produce the final map. The built infrastructure in two sites on the North Slope of Alaska are mapped by Ref. 19 using DL approach and 4-band commercial satellite images with resolutions from 0.5 up to 0.87. The utilized model in this work is the U-Net with ResNet50 as the backbone. An encoder–decoder-based network by dilated convolutions is proposed by Ref. 20 for segmenting building rooftops in RGB RS images.

In this study, we propose a postprocessing process after segmenting reservoirs to detect errors and construct an accurate reservoir map. Furthermore, man-made objects in both urban and countryside areas are studied. We also suggest a method to detect RoIaR using image processing techniques. In the proposed two-phase workflow, we initially detect the reservoirs, followed by identifying the RoI, and ultimately segmenting man-made objects within the RoI. In this way, we avoid annotating and predicting areas outside the RoI. These are the main relevant contributions of this paper.

Although elevation data can improve the detection process, they are not currently viewed as a cost-effective solution to map RS images.[21] Moreover, spatial resolution is more critical than spectral resolution in urban land cover mapping.[9] Therefore, we collected the data using the Google Earth (GE) platform, which is a widely used database.[22,23] In this platform, we have access to free high-resolution RS images from target reservoirs at various times. GE covers more than 25% of the Earth's land surface and three-quarters of the global population by images with submeter resolution.[24,25] Therefore, it can be used for studying many other reservoirs.

The organization of our paper is as follows. Sec. 2 describes the studied reservoirs, collected data characteristics, applied data preprocessing pipeline, the proposed workflow for segmenting man-made objects around the reservoirs, and corresponding utilized methods. Next, the performance of each workflow stage, besides results visualization and workflow evaluation, is explored in Sec. 3. The results and findings of the study are discussed in Sec. 4. Finally, this paper is concluded in Sec. 5.

## 2 Materials and Methods

Our task of interest is man-made object segmentation around reservoirs. The proposed approach (see Fig. 1) is based on three main steps: (1) reservoir segmentation; (2) RoIaR extraction; and (3) man-made object segmentation in the RoIaR. The data are initially collected and preprocessed to be prepared in a suitable manner. The input images are processed in phase-1 for reservoir segmentation. The resulting reservoir map is then forwarded to phase-2, where the RoIaR is detected. This RoIaR serves as a mask for the final segmentation of man-made objects. Details of the proposed workflow and the steps implemented for its preparation are provided in the following sections.

### 2.1 Data Collection

Our experiments are performed on RGB RS images collected from eight reservoirs in Brazil using the Google Earth Pro© software. GE images represent an integration of multiple satellite data sources, mainly DigitalGlobe's QuickBird commercial satellite and EarthSat.[26] Aiming at improving the appearance of the images, the spectral information of images with more than three bands is reduced to RGB.[27] Furthermore, the appearance of GE images is improved using color balancing, warping, and mosaic processing.[3] The GE platform presents important advantages, such as the fact of being an open database of RS images, of including historical images and of the flexibility in selecting images of different resolutions.

The eight studied reservoirs are Anta, Billings (the largest reservoir in São Paulo, Brazil), Dona Francisca, Guarapiranga, Jaguara, Luiz Barreto, Nova Avanhandav (Nova), and Salto Osório. Their geographic coordinates are listed in Table 1, and their locations are visualized in Fig. 2. For each reservoir, images over two different years are collected (Table 2). A total of 206 images, each with $2683 \times 4800$ pixels are obtained, encompassing varying view altitudes and thus resulting in different resolutions (from $\sim$1 up to 2 m).

### 2.2 Data Preparation and Annotation

Data preparation involves two aspects: preprocessing for mosaic image formation and data annotation. The data preparation scheme is illustrated in Fig. 3 using Guarapiranga reservoir samples. The data preparation aims to prepare data for training the phase-1 and phase-2 segmentation models in Fig. 1.

As shown in Fig. 3, the input images are initially mosaicked to eliminate overlapping areas in collected GE images. Subsequently, the images are annotated into two classes: reservoir and non-reservoir. Constructing the mosaic images is also essential for implementing the next steps.

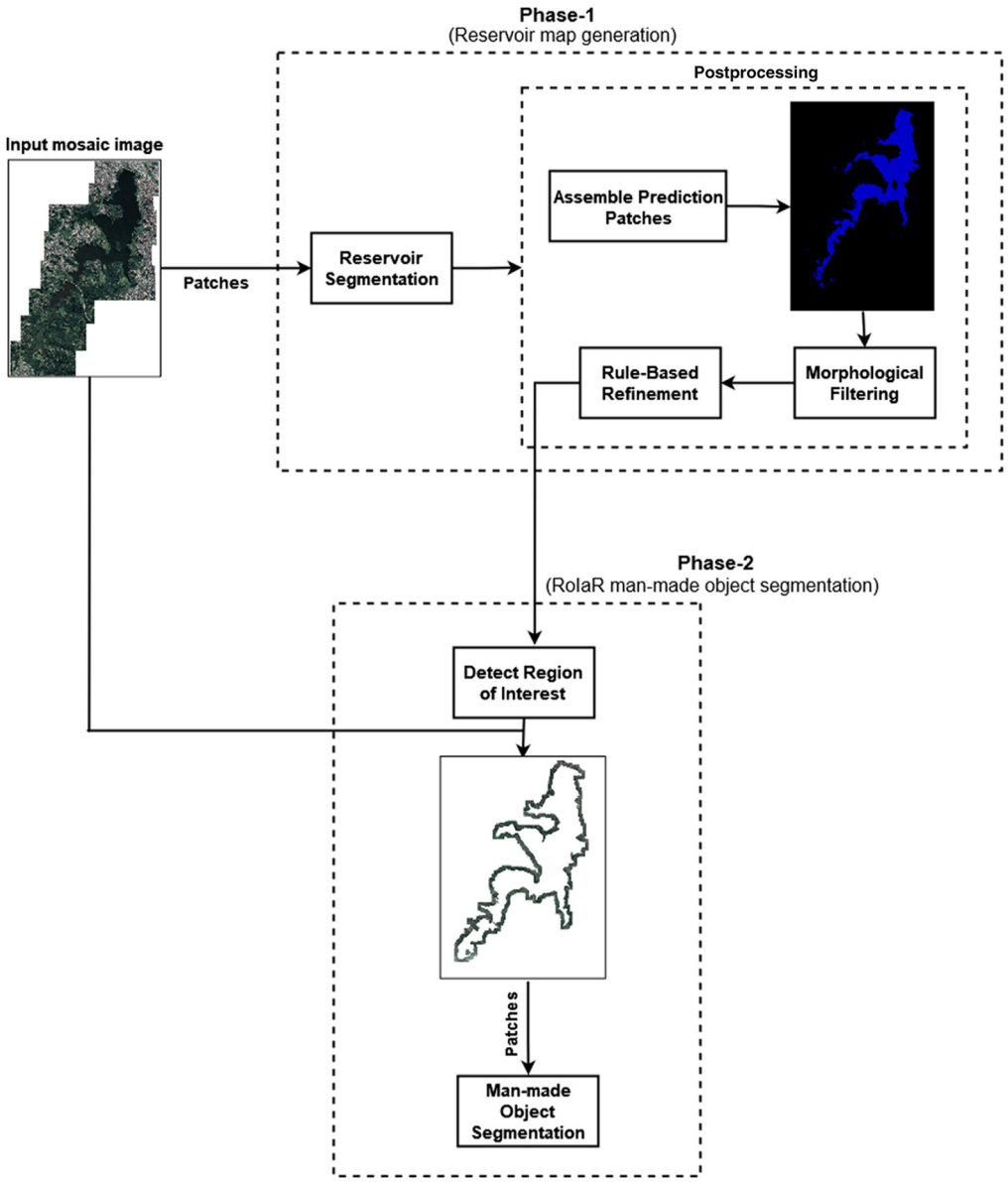


**Fig. 1** Overview of the proposed workflow.

**Table 1** Locations of the studied reservoirs.

| Reservoir | State | Coordinates |
|---|---|---|
| Anta | Minas Gerais and Rio de Janeiro | 22°02′33.20′′ S, 43°01′16.85″ W |
| Billings | São Paulo | 23°48′50.62″ S, 46°32′19.39″ W |
| Dona Francisca | Rio Grande do Sul | 29°26′34.18″ S, 53°16′09.09″ W |
| Guarapiranga | São Paulo | 23°43′16.93″ S, 46°44′22.23″ W |
| Jaguara | Minas Gerais and São Paulo | 20°05′01.85″ S, 47°24′10.44″ W |
| Luiz Barreto | São Paulo | 20°14′18.50″ S, 47°11′01.95″ W |
| Nova | São Paulo | 21°10′34.54″ S, 50°07′34.03″ W |
| Salto Osório | Paraná | 25°33′28.60″ S, 52°57′07.61″ W |


**Fig. 2** Visualization of the studied reservoirs locations.

**Table 2** Acquisition years of each reservoir dataset. Some of the older year images of Luiz and Nova belong to 2004 and 2010, respectively.

| | Acquisition Years | |
|---|---|---|
| Reservoir | Older | Earlier |
| Anta | 2014 | 2020 |
| Billings | 2009 | 2019 |
| Dona Francisca | 2011 | 2017 |
| Guarapiranga | 2009 | 2019 |
| Jaguara | 2010 | 2020 |
| Luiz Barreto | 2010 | 2020 |
| Nova | 2011 | 2021 |
| Salto Osório | 2005 | 2019 |

Next, in order to simplify the contour around the reservoirs, a polygonal approximation is initially carried out.[28–30] This allows controlling the coarseness by the polygonal approximation parameter. Then a rectangular box connecting each pair of consecutive polygon corners is defined. These boxes are enlarged to cover a minimum distance from the border of the reservoir, which is set empirically. The RoIaR is defined as the union of these boxes (see Fig. 3) and used to mask the mosaic image.

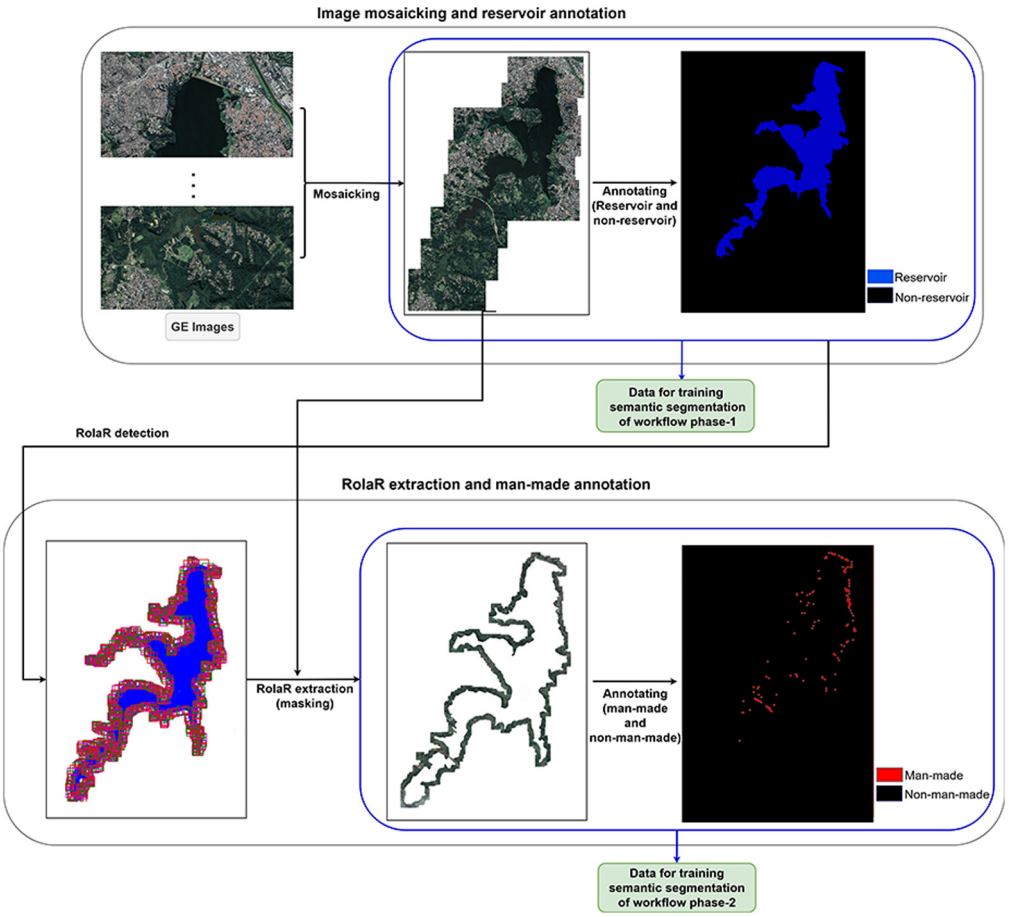


**Fig. 3** Proposed data preparation and annotation pipeline.

The masked RoIaR image is annotated into man-made and non-man-made objects:

- *man-made objects*: road (asphalted and not-asphalted), rooftop, bridge, pool, urban, and countryside constructions, impervious surface.
- *non-man-made objects*: vegetation, water body, bare land, etc.

### 2.3 Phase-1: Reservoir Map Generation

**Reservoir segmentation:** This step explores a deep neural network that segments input RGB patches into reservoir and non-reservoir. In this process, we trained and compared two encoder–decoder-based models, the U-Net, and SegNet architectures. Following our evaluation, the SegNet-based model emerged as the most effective choice in our experiments. Below, we briefly describe these two architectures.

The U-Net architecture introduced by Ref. 31 is based on a downsampling–upsampling procedure that concatenates feature maps between each encoder and corresponding decoder by skip connections (see Fig. 4). In each step in the encoder path, two $3 \times 3$ convolutions followed by a ReLU and a $2 \times 2$ max-pooling with stride two are repeated. Furthermore, the number of feature channels in each downsampling step is doubled. After each upsampling in the decoder path, a $2 \times 2$ convolution that halves the number of feature channels is applied. These features are concatenated with the cropped feature of the corresponding encoder step, and then two $3 \times 3$ convolution-ReLU blocks are implemented.

Due to the unpadded convolutions utilized in the U-Net, the output size of the model is smaller than the input. Therefore, we avoided unpadded convolutions to keep the size of each output equal to the corresponding input (named U-Net_p). On the other hand, a common strategy in DL research for training the CNNs properly and avoiding training from scratch is to use a

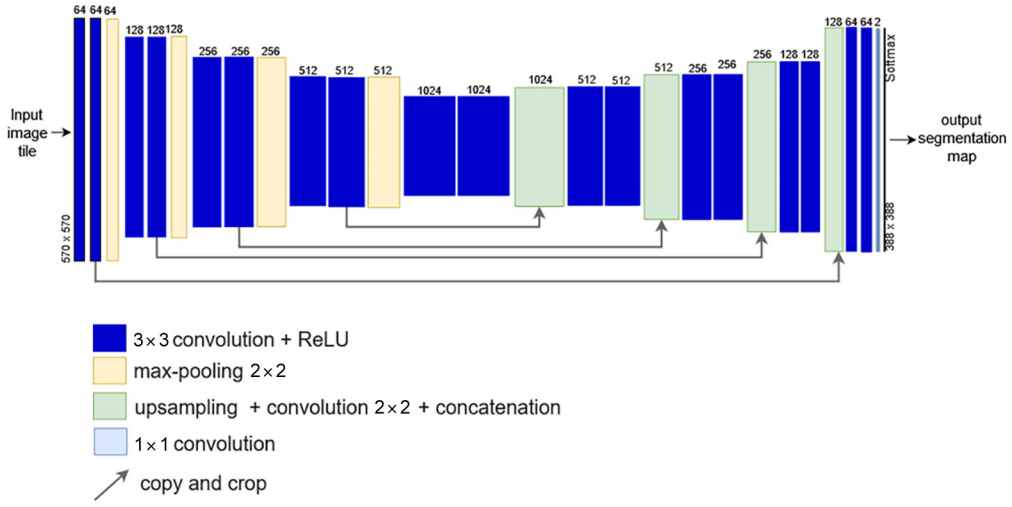


**Fig. 4** U-Net architecture.

pretrained CNN as the initializer or as the fixed feature extractor, called transfer learning. Therefore, we trained a U-Net model whose encoder path was replaced by VGG-16 (named U-Net_v) and initialized that with weights trained on the ImageNet dataset. However, the model overfitted the training set. The last trained U-Net-based model (named U-Net_s) has fewer features. In this model, there is only one convolution block in each layer that is also batch normalized.[32]

The SegNet architecture was first introduced by Ref. 33. Similar to the U-Net, SegNet includes an encoder and a decoder part with the advantage that the need for learning to upsample is eliminated. Since each decoder uses pooling indices computed in the max-pooling step of the corresponding encoder. After each convolution layer in the encoder path, a ReLU non-linearity is used, whereas, in the decoder, no ReLU non-linearity is presented. Furthermore, the number of channels per layer is constant (see Fig. 5). In the employed architecture (called SegNet_d), despite the original form, the number of feature channels is doubled at each downsampling step. Batch normalization is applied after each convolution layer.

The collected images correspond to different reservoirs geographically spread in Brazil and have different visual properties. They may be obtained in different seasons, atmospheric conditions, geological conditions, and so forth. A possible approach to address such variability is to adopt domain adaptation techniques. Since this is out of the scope of this paper, we explored a data splitting approach to ensure variability in the training, validation, and test sets. Samples from every mosaic image are used in these sets in the following proportions: 60% for the training set, 20% for the validation set, and 20% for the test set.

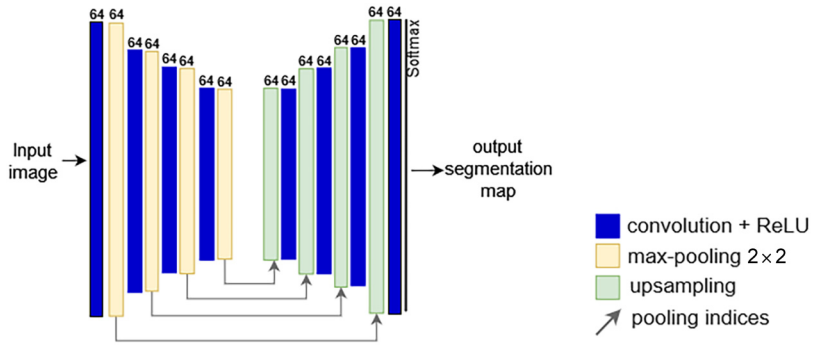


**Fig. 5** SegNet architecture.

**Postprocessing:** Feeding models by mosaic images instead of patches are impossible because of the available GPU memory limits. In many cases, patches do not contain important information about objects, such as their shapes, sizes, and locations in the images. However, this information is essential for detecting some water bodies from reservoirs. On the other hand, spectral similarities between objects of different classes also cause errors. Therefore, we proposed a postprocessing stage to fix these errors.

In this stage, the segmented patches are initially assembled to form the reservoir map. The morphological opening is applied to remove small false positive pixels and other non-interesting water objects that are segmented partially or completely as reservoir objects. Next, morphological closing is applied to remove small false negative objects inside the reservoir objects.

Applying morphological transformations with a large kernel causes changes in the shapes of objects predicted as the reservoir. In order to remove errors inside reservoirs (such as floating vegetation) and noisy objects (such as large water bodies around reservoirs), without removing reservoir objects that are separated because of constructed bridges, the following rules are proposed.

- If a non-reservoir object is surrounded by a reservoir object, it is classified as the reservoir.
- If the size of a reservoir object is smaller than one-tenth of the size of the largest reservoir object, or the minimum distance between these two objects is >300 m, then it is classified as non-reservoir.

### 2.4 Phase-2: RoIaR Man-Made Object Segmentation

**RoIaR extraction:** Once the reservoir is segmented, the next step is to detect and extract the RoIaR. Two possible approaches for RoIaR detection have been considered: polygonal approximation-based and mathematical morphology-based. The polygonal approximation approach has been described in Sec. 2.2, which is the one adopted for dataset annotation. Although this approach is useful for sparse data annotation (because we may control the polygonal approximation parameters), it produces patches of varying sizes that may not be suitable for analyzing man-made objects' evolution, for instance.

Therefore, a mathematical morphological approach is also explored. Let $I$ denotes the segmented reservoir object and $s$ a structuring element. The dilated reservoir object is defined as $I_d = I \oplus s$, where $\oplus$ is the morphological dilation. The RoIaR $R$ is defined as $R = I_d - I$, where $-$ denotes set difference.

Following the data annotation procedure illustrated in Fig. 3, the detected RoIaR is applied as a mask to the mosaic image for RoIaR extraction. The extracted RoIaR is then segmented into man-made and non-man-made objects.

**Man-made object segmentation:** Two widely used network architectures for RS segmentation are the pyramid networks and encoder–decoder networks.[34] In phase-2, we assessed the PSPNet, FPN, and LinkNet networks, whose details are presented in the next paragraphs.

The PSPNet has been introduced by Ref. 11 and won the ImageNet Scene Parsing Challenge 2016. It is a pyramid pooling module that enables the network to capture the context of the whole image. In this module, the feature map is pooled at different sizes and passed through a convolution layer. Next, these features are upsampled and concatenated with the original feature map and passed through a convolution layer to produce the final prediction (see Fig. 6). We implemented PSPNet with different backbones in this study.

Figure 7 presents the FPN's general schema, a network initially proposed by Ref. 35 for object detection. The construction of this architecture involves a bottom-up path, a top-down path, and lateral connections. The scaling step in the bottom-up path (and consequently in the top-down path) is two. Each lateral link combines feature maps from the bottom-up and top-down pathways with the same spatial size. Finally, the feature maps in the top-down stages are upsampled to be the same size as the input image. These feature maps are combined and used to produce the prediction map. The ResNet is used as the backbone, whereas in this study, other backbones have also been experimented.

The LinkNet architecture proposed by Ref. 36 is a fast semantic segmentation method that is constructed from an encoder and a decoder path (see Fig. 8). Each residual block in the encoder path consists of two consequent convolution blocks. The input of each residual block is bypassed

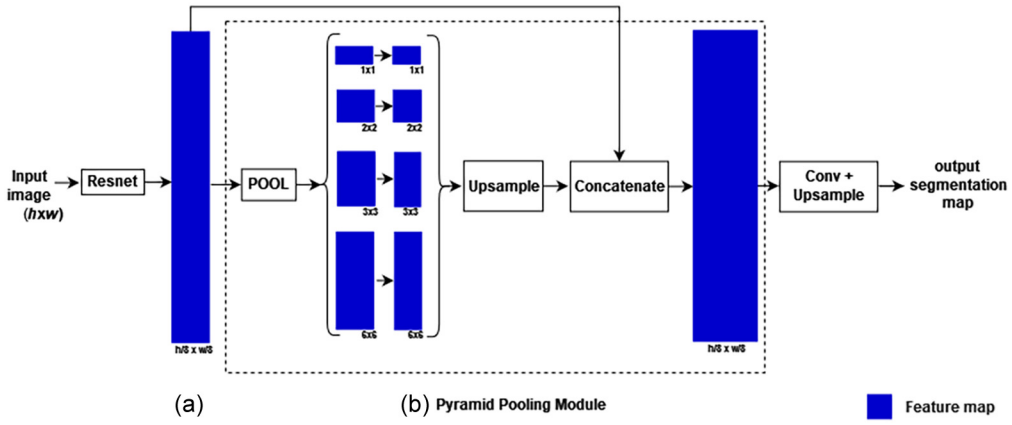


**Fig. 6** An overview of PSPNet. The size of feature map channels is denoted below each box. The size of the last feature map in (a) is 1/8 of the input image size.

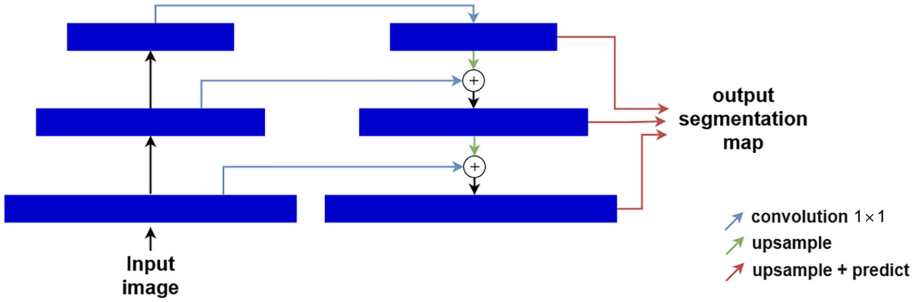


**Fig. 7** Overview of FPN.

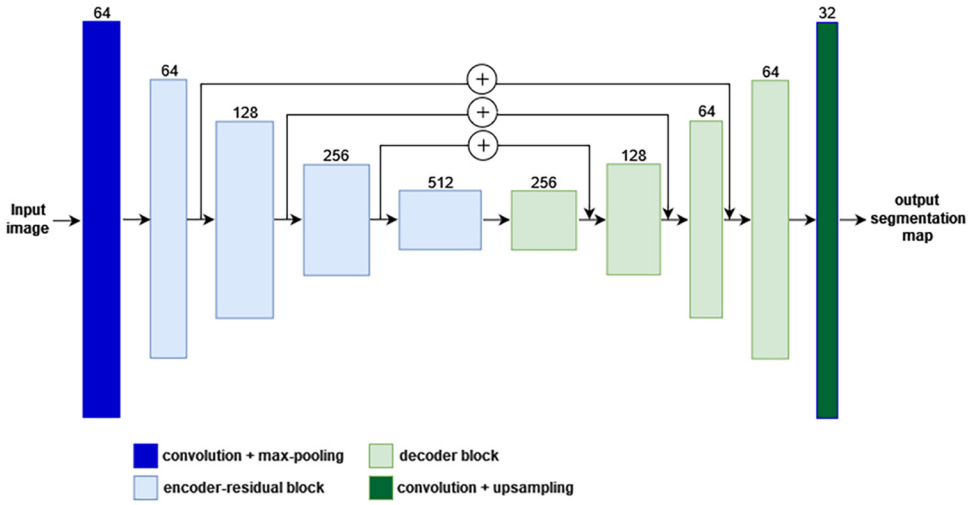


**Fig. 8** Overview of LinkNet architecture.

to its output. The decoder blocks consist of three convolution layers, and the middle is a full convolution. The advantage of the proposed architecture is passing the input of each encoder block to the output of the corresponding decoder block.

Splitting data into the training and test sets are reported to work well when the dataset size is modest. On the other hand, the training and test sets must represent possible distributions of the addressed problem. Therefore, 70% for the training set and 30% for the test set are selected randomly from each RoIaR.

**Loss function:** The focal loss proposed by Ref. 37 and the Dice loss (DL)[38] are utilized as the loss functions for training the networks. Focal loss down-weights easy examples and hence helps the model to learn complex examples better. It is reported by Ref. 39 that focal loss works best when the data is highly imbalanced. To see how it works, first, consider the binary cross entropy loss (CE):

$$\mathrm{CE}(p,y)=\begin{cases}-\log\ p, & \text{if } y=1\\ -\log(p-1), & \text{otherwise}\end{cases}, \tag{1}$$

where $p$ is the predicted probability for class with label $y=1$. Now let define a new notation $p_t$:

$$p_t=\begin{cases}p & \text{if } y=1\\ p-1, & \text{otherwise}\end{cases}. \tag{2}$$

Using this notation, we can rewrite Eq. (1) as $\mathrm{CE}(p_t)=-\log(p_t)$. To balance the importance of positive/negative examples, we can consider $\alpha_t$ as the weight for class 1 and $1-\alpha_t$ for class 0, then $\alpha$-balanced CE will be written as

$$\mathrm{CE}(p_t)=-\alpha_t\log(p_t). \tag{3}$$

Finally, to down-weight easy examples, they add factor $(1-p_t)^\gamma$ to CE where $\gamma>0$ is a tunable parameter. Based on the experiment, $\gamma=2$ works best and is used in this study too.

The DL is based on the dice coefficient (DC) [see Eq. (4)]. In the case of binary classification, $A$ is the set of all positive examples, and $B$ is the set of correct predicted positive examples:

$$\mathrm{DC}=\frac{2|A\bigcap B|}{|A|+|B|}. \tag{4}$$

Then DC can be expressed as the following form:

$$\mathrm{DC}=2\cdot\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}, \tag{5}$$

where TP, FP, and FN are true positive, false positive, and false negative, respectively. The DL takes the following form:

$$\mathrm{DL}=1-2\cdot\frac{\sum_{i=1}^{N}p_i r_i}{\sum_{i=1}^{N}r_i+p_i}, \tag{6}$$

where $p_i$ is the predicted probability for pixel $i$'th and $r_i$ is the ground truth of the corresponding pixel. The imbalance between the foreground and background can be efficiently reduced using DL. However, it disregards the imbalance in data difficulty.

## 3 Experimental Results

This section describes the experimental evaluation of the proposed workflow. Phases 1 and 2 have been evaluated, and the results are discussed below. We have explored some open-source libraries[40,41] in our code for developing the DL architectures.

### 3.1 Performance Evaluation Metrics

We adopted three common statistics, precision [Eq. (7)], recall [Eq. (8)], and $F_1$-score [Eq. (9)], as well as the confusion matrix of segmentation maps:

$$\text{precision}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \tag{7}$$

$$\text{recall}=\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \tag{8}$$

$$F_1=2\cdot\frac{\text{precision}\times\text{recall}}{\text{precision}+\text{recall}}. \tag{9}$$

**Table 3** Performance of trained architectures for phase-1 segmentation stage.

| | $F_1$-score | |
|---|---|---|
| Model | Training set | Validation set |
| U-Net_p | 96.19 | 95.46 |
| U-Net_v | 92.11 | 68.86 |
| U-Net_s | 98.16 | 97.80 |
| SegNet_d | 98.40 | 98.03 |

### 3.2 Phase-1 Experimental Results

The trained architectures for this phase are modified versions of U-Net and SegNet. All trained models use the binary cross entropy as the loss function. The learning rate in the Adam optimizer (proposed by Ref. 42) is set to 0.001, which is reduced by a factor of 0.2 after every five epochs with no reduction in validation loss down to $10^{-7}$. Although the number of epochs is set to 100, training is stopped after 20 epochs with no reduction in the validation loss. Patches with $416 \times 608$ pixel sizes are fed into the networks, and training, validation, and test sets contain 6017, 2009, and 1998 patches, respectively. Vertical and horizontal flips are two types of data augmentation that each one is applied randomly on 50% of training set patches. The $F_1$-score of trained models in segmenting the training and validation sets are presented in Table 3. As is illustrated in this table, the U-Net_v overfits the training set.

Table 4 presents the performance of models with healthy learning curves in the segmentation of the validation set, and it is worth noting that SegNet_d outperforms the U-Net-based models. The performance of SegNet_d in segmenting the test set is illustrated in Table 5. Some patches of studied reservoirs with different spectral properties besides their ground truths and SegNet_d, U-Net_s, and U-Net_p prediction outputs are shown in Fig. 9.

In addition to errors that occur because of spectral similarities between reservoirs and some other objects (such as shadows), there are small water bodies in the images that are segmented as the reservoir by the models. This issue is unavoidable because of feeding patches to the models instead of the original images. Therefore, postprocessing the network outputs is an essential task. Figure 10 presents examples of non-interesting water bodies in the collected dataset and their segmentation results in the generated reservoir maps.

**Table 4** Performance of models with healthy learning curves for phase-1 segmentation stage on the validation set.

| | Precision | | Recall | | $F_1$-score | |
|---|---|---|---|---|---|---|
| Model | Non-reservoir | reservoir | Non-reservoir | Reservoir | Non-reservoir | Reservoir |
| U-Net_p | 98.18 | 93.72 | 98.71 | 91.27 | 98.44 | 92.48 |
| U-Net_s | 98.63 | 93.85 | 98.71 | 93.49 | 98.67 | 93.67 |
| SegNet_d | 98.79 | 94.39 | 98.82 | 94.24 | 98.81 | 94.32 |

**Table 5** SegNet_d performance in segmenting the test set.

| Class | Precision | Recall | $F_1$-score | Support (No. pixels) |
|---|---|---|---|---|
| Non-reservoir | 98.82 | 98.87 | 98.85 | 4,211,777,759 |
| Reservoir | 94.33 | 94.11 | 94.22 | 84,172,385 |

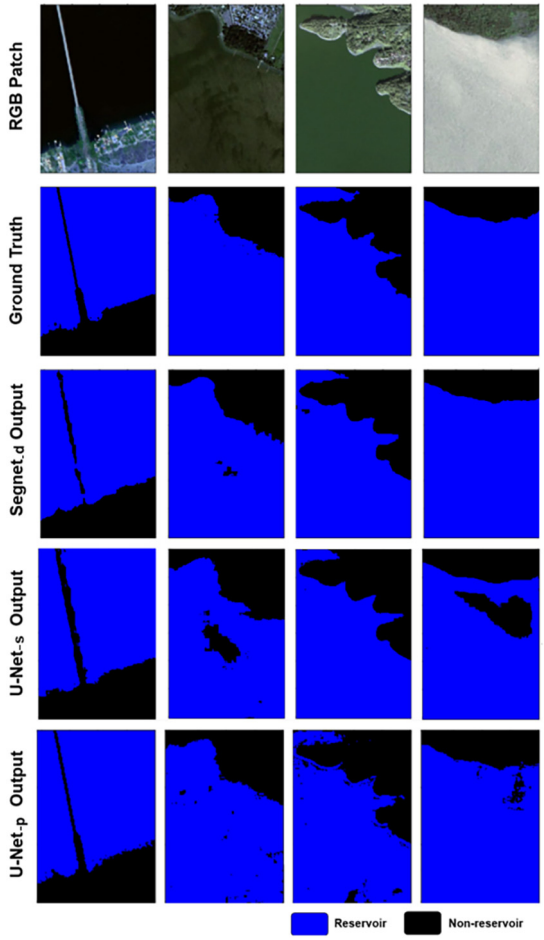


**Fig. 9** Examples of the test set patches beside their corresponding ground truths and segmentation outputs.


**Fig. 10** Examples of non-reservoir water bodies in the collected dataset and corresponding segmentation results.

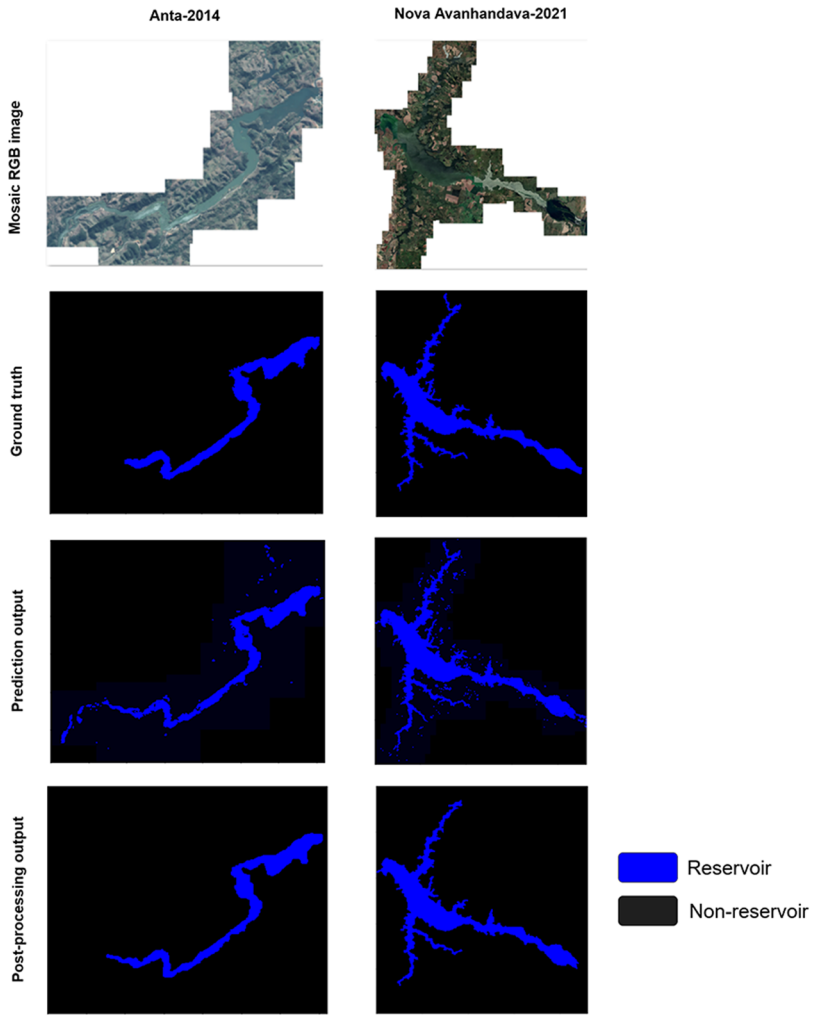


**Fig. 11** Two examples of produced mosaic images, corresponding ground truths, prediction outputs, and postprocessing outputs. Anta-2014 with 11,687 × 14,430 pixel size, and Nova-2021 with 24,830 × 23,193 pixel size are depicted in the first and second columns, respectively.

Morphological operations in postprocessing are highly effective in removing minor errors, as described above. The applied structuring element size for each reservoir is set to 100 divided by the spatial resolution. For instance, if the spatial resolution of a mosaic image is 1 m, the structuring element size is $100 \times 100$.

As the reservoirs contain branches, applying morphological operations with large kernel sizes increases FP and FN objects. Accordingly, significant errors are removed by applying the two rules to objects in the produced segmentation maps. Postprocessing solely using rules is time-consuming because of the high number of FP and FN objects in prediction maps, whereas morphological operations speed up this process. Anta-2014 and Nova-2021 mosaic images, besides their ground truths, model outputs, and postprocessing outputs are illustrated in Fig. 11. The SegNet_d performance in segmenting these two reservoirs besides postprocessing performance are presented in Table 6. Applying the proposed postprocessing improves the accuracy of produced reservoir maps except for two of the 16 studied cases.

### 3.3 Phase-2 Experimental Results

As discussed above, VGG-16, ResNet-50, and ResNet-101 are the most frequented backbones.[7] In this study, these three backbones besides EfficientNet-B4 have been experimented. All backbones are initialized with weights trained on the ImageNet dataset. The Adam optimizer is used as the optimizer in all models. The initial learning rate is set to 0.0001 or 0.001, which is automatically reduced by a factor of 0.2 after every five epochs with no reduction in validation loss

**Table 6** Prediction and refinement performance metrics for Anta-2014 an Nova-2021.

| | | Model | | Postprocessing | |
|---|---|---|---|---|---|
| Reservoir | Class | Precision | Recall | Precision | Recall |
| Anta-2014 | Non-reservoir | 98.45 | 98.95 | 99.15 | 99.56 |
| | Reservoir | 89.75 | 85.53 | 95.72 | 92.07 |
| Nova-2021 | Non-reservoir | 98.33 | 98.03 | 98.67 | 99.13 |
| | Reservoir | 92.56 | 93.65 | 96.63 | 94.90 |

down to $10^{-7}$. The mini-batch size is set to two and power of two (up to the possible size based on the model's size and available memory). The number of epochs in training all models is set to 80. The vertical and horizontal flips are two data augmentation methods that are implemented on different portions of images (up to 0.7). We added dropout regularization (<0.3) to the models with overfitting. Furthermore, experiments are on two training sets, a training set containing 70% of data or the oversampled training set. The oversampled images are images with at least 200 man-made objects pixels.

The evaluation metrics for the highest performance model constructed using each architecture are presented in Table 7. These models are all trained on the oversampled training set with a learning rate of 0.0001. Furthermore, the data augmentation rate in these models is set to 0.7 for each data augmentation method, and the dropout regularization is set to 0.0, 0.3, 0.3, and 0.0, respectively. Regarding the $F_1$-score, the best performance belongs to FPN; however, the differences are insignificant. The utilized backbones for each model in this table are ResNet50, VGG-16, VGG-16, and Efficientnet-B4, respectively. Except for the PSPNet that VGG-16 could improve the performance of the model significantly (2.34%), the performances of the rest models are slightly affected by changing their backbones (<0.73%). In our experiments, oversampling images with more than 200 man-made object pixels improved the performances. Despite the expectation, increasing batch size did not increase the performance metrics in all cases. Adding the DL to the focal loss function significantly improved the models' performances. Although increasing the data augmentation rate prevented overfitting in some cases, in other cases increasing dropout and data augmentation rates were both essential. Though the FPN outperforms the PSPNet, each epoch training time of PSPNet is less than one-third of the FPN. FPN performance in segmenting test set is presented in Table 8.

Moreover, the FPN performance in segmenting RoI of reservoirs located in the countryside and urban areas are computed separately and shown in Table 9. Some examples of patches besides their ground truths and segmentation outputs are illustrated in Fig. 12. This figure illustrates examples of different types of roads, rooftops, and urban and countryside constructions with different density levels.

**Table 7** Highest achieved performances using trained models for phase-2 segmentation on training and test sets.

| | $F_1$-score | |
|---|---|---|
| Model | Training set | Test set |
| U-Net | 91.64 | 90.13 |
| PSPNet | 91.39 | 89.76 |
| FPN | 92.16 | 90.32 |
| LinkNet | 91.95 | 90.15 |

**Table 8** FPN performance in segmenting test set into the man-made and non-man-made objects pixels.

| Class | Precision | Recall | $F_1$-score | Support (No. pixels) |
|---|---|---|---|---|
| Non-man-made | 99.52 | 99.56 | 99.54 | 327,065,669 |
| Man-made | 81.79 | 80.43 | 81.10 | 8,101,819 |

**Table 9** FPN performance in segmenting countryside and urban man-made objects. C and U are the abbreviations for countryside and urban.

| | Precision | | Recall | | $F_1$-score | |
|---|---|---|---|---|---|---|
| Class | C | U | C | U | C | U |
| Non-man-made | 99.68 | 99.39 | 99.73 | 99.26 | 99.71 | 99.33 |
| Man-made | 78.70 | 86.62 | 75.78 | 88.75 | 77.21 | 87.67 |

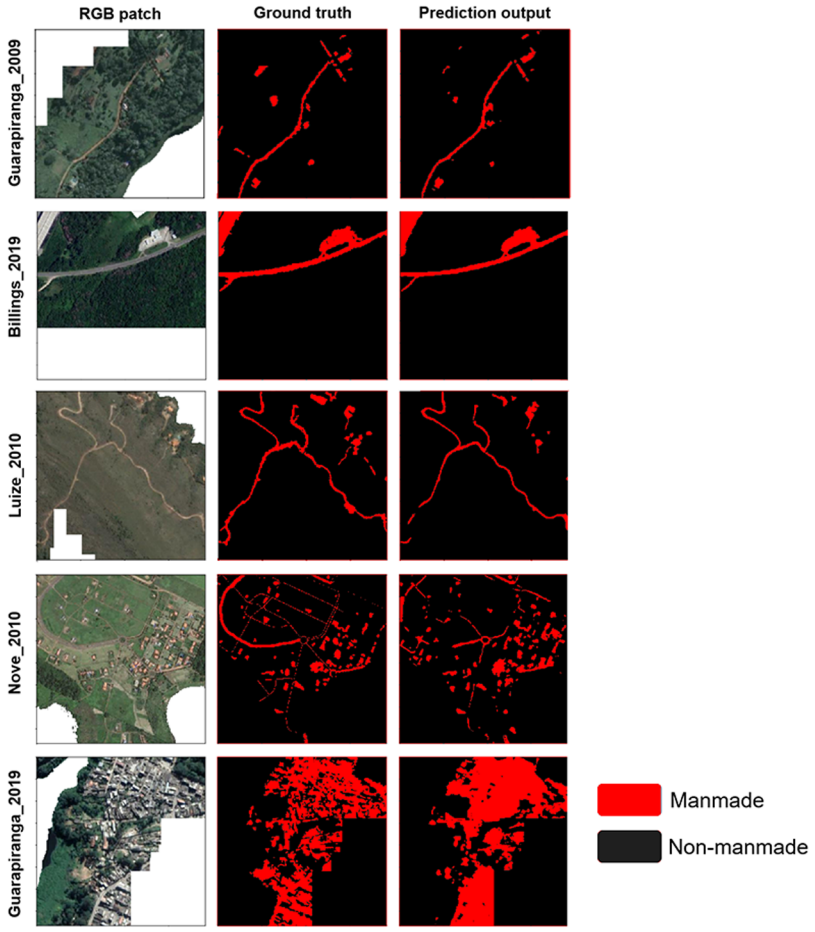


**Fig. 12** Examples of studied reservoirs RoI patches beside their corresponding ground truths and prediction outputs.

### 3.4 Workflow Evaluation

We evaluated the proposed workflow using a dataset collected from the Barra Grande reservoir (Barra). Barra is located in Santa Catarina and Rio Grande do Sul states in Brazil. The collected images belong to 2021, and their spatial resolution is two meters. To evaluate the proposed

**Table 10** Performance of phase-1 segmentation and postprocessing stages in segmenting Barra dataset to reservoir and non-reservoir.

| | Model | | | Postprocessing | | |
|---|---|---|---|---|---|---|
| Class | Precision | Recall | $F_1$-score | Precision | Recall | $F_1$-score |
| Non-reservoir | 98.39 | 96.86 | 97.62 | 98.38 | 98.36 | 98.37 |
| Reservoir | 84.00 | 91.21 | 87.45 | 90.92 | 91.04 | 90.98 |

workflow using the collected data, first, patches with $416 \times 608$ pixel size are constructed from the mosaic RGB image of Barra. Next, patches are fed to the trained SegNet_d to be segmented into the reservoir and non-reservoir. The SegNet_d performance is evaluated by comparing model outputs with manually produced ground truths. In the next step, the SegNet_d outputs are assembled to be refined using the proposed postprocessing stage. The refined reservoir map is used to detach the RoI around Barra. The covered distance from the border of the reservoir is 200 meters. In Table 10, the performances of the phase-1 segmentation stage, besides the performance of proposed postprocessing, are reported. Table 11 shows the evaluation metrics for the phase-2 segmentation stage. Furthermore, some samples of phase-1 and phase-2 segmentation outputs are illustrated in Figs. 13 and 14, respectively.

**Table 11** Performance of phase-2 segmentation stage in segmenting Barra RoI to man-made and non-man-made.

| Class | Precision | Recall | $F_1$-score |
|---|---|---|---|
| Non-man-made | 99.99 | 99.99 | 99.99 |
| Man-made | 73.29 | 79.43 | 76.23 |

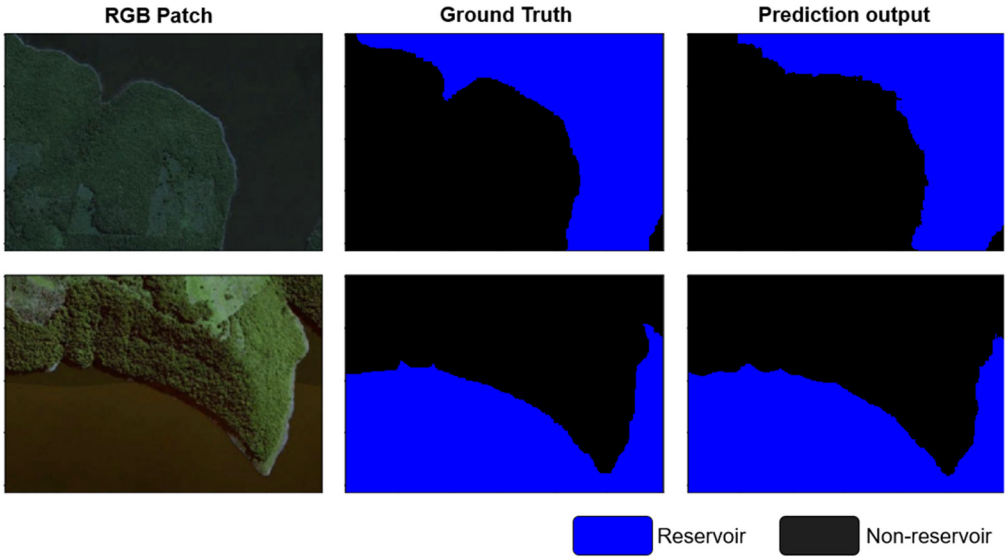


**Fig. 13** Two samples of Barra phase-1 patches, besides their corresponding ground truths and reservoir segmentation results.

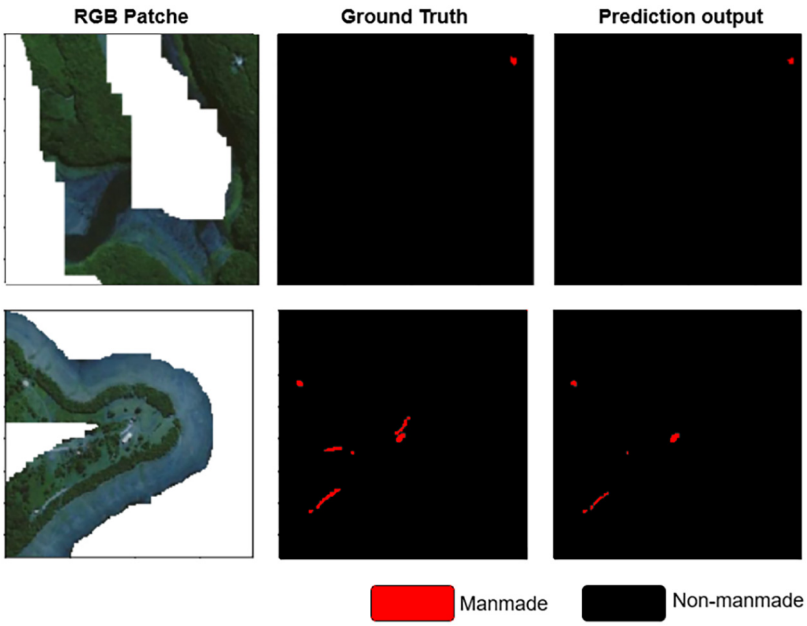


**Fig. 14** Two samples of Barra RoI patches, their corresponding ground truths, and man-made object segmentation results.

**Table 12** U-Net performance in segmenting the training and test sets into reservoir, man-made, and non.

| | Precision | | Recall | | $F_1$-score | |
|---|---|---|---|---|---|---|
| Class | Train | Test | Train | Test | Train | Test |
| Reservoir | 96.58 | 96.15 | 96.52 | 95.72 | 96.55 | 95.94 |
| Man-made | 62.17 | 59.02 | 50.78 | 49.13 | 55.90 | 53.62 |
| Non | 98.64 | 98.37 | 98.88 | 98.69 | 98.76 | 98.53 |

### 3.5 Benchmark

In order to show the effectiveness of our proposed two-phase approach, we applied a single-phase network to segment images into reservoir, man-made, and non-man-made, as the baseline. In this model, the VGG-16 is used as the backbone, the learning rate is set to 0.0001, the number of epochs is set to 150, the early stopping is not applied, and the summation of Dice and focal losses is used as the loss function. The learning rate is reduced by a factor of 0.2 after every five epochs with no reduction in validation loss down to $10^{-7}$. Same as the phase-2 training phase, we constructed patches with $384 \times 384$ pixel size and split them into two sets, training and test.

Since man-made objects inside RoIaR are annotated as man-made and outside as non-man-made (because they are not around the reservoir), the baseline performance is poor (see Table 12), as expected. This simple baseline approach illustrates the importance of our proposed two-phase approach.

## 4 Discussion

The experimental performance evaluation has addressed the results of phases 1 and 2 of the proposed workflow, workflow validation by an external testing dataset, and the single-phase segmentation benchmark result.

Reservoir segmentation is addressed in phase-1 of the workflow. We trained three U-Net-based models in this phase. The vanilla U-Net was changed to keep the size of each output equal to the corresponding input to produce a pixelwise classification. In addition, a U-Net with

VGG-16 as the backbone was trained. The model over-fitted highly to the training set. Decreasing the number of feature maps in the model (named U-Net_s) caused performance improvement and fixed the overfitting issue, as shown in Table 3. A SegNet-based architecture was also trained to examine its ability to enhance segmentation outputs. However, it outperformed the U-Net_s slightly (1.23% in $F_1$-score).

In the DL segmentation studies, the reservoirs are considered in a broad class called water bodies. In this study, a postprocessing stage is proposed to eliminate errors caused by floating vegetation and delete FP and FN objects caused by spectral similarities between reservoirs and other objects. The proposed postprocessing improved the overall accuracy and provided a clear map of the reservoirs, as shown by the examples in Table 6 and Fig. 11.

Phase-2 restricts the segmentation of man-made objects in the RoIaR. Four DL architectures have been evaluated to segment the man-made objects: U-Net, FPN, LinkNet, and PSPNet. This problem typically involves imbalanced data because of government policies to protect such areas besides difficulty in segmenting countryside man-made objects.

In order to address these issues, we tried out the capability of two recommended loss functions (dice and focal losses) and the oversampling strategy. Although focal loss was reported as the best loss function for segmenting unbalance data, adding DL to the focal loss significantly improved the performances. Furthermore, oversampling improved the performances as well. We trained each architecture with four different backbones, ResNet50, ResNet101, VGG-16, and EfficientNet-B4. The highest improvement caused by changing the backbone belongs to VGG-16 in PSPNet, 2.34%, whereas changing the backbone in other architectures had a low contribution.

Workflow validation has been carried out using data not seen by the model during training (Barra reservoir, see Sec. 3.4). The validation data included realistic noise and difficulties, such as clouds. Despite this, the phase-1 model achieved to 92.54% average $F_1$-score that was even improved to 94.68% by applying postprocessing techniques (see Table 10). Additionally, the reservoir is in the countryside. The majority of roads are not asphalted, and man-made objects present different visual features from urban areas. Also there are fewer samples of them in the training data. Accordingly, segmenting them is more complicated compared to urban areas. Nonetheless, the phase-2 model could gain an acceptable performance, as seen in Table 11.

In order to show the effectiveness of the proposed two-phase approach compared to a single-phase approach, we trained a network to segment images into reservoir, man-made, and non-man-made. We increased the feature maps in the phase-2 trained U-Net-based model, the VGG-16 is used as the backbone, the learning rate is set to 0.0001, the number of epochs is set to 150, and the summation of dice and focal losses is used as the loss function. The learning rate is reduced by a factor of 0.2 after every five epochs with no reduction in validation loss down to $10^{-7}$. The data were split into the training and test sets, and no early stopping was applied. Nonetheless, the model was poor in segmenting man-made objects (see Table 12), and its man-made $F_1$-score is 35.74 % less than the phase-2 U-Net-based model.

## 5 Conclusions

In this study, we proposed a two-phase workflow to segment man-made objects around reservoirs in an end-to-end procedure. In order to improve produced reservoir maps, a postprocessing stage is proposed that, besides increasing the precision metric, its effect is remarkable by visual evaluation. A small portion of images belongs to the class of man-made object, specially countryside man-made object. Nonetheless, we gained promising results by collecting images of reservoirs mainly located in the countrysides, and defining a suitable loss function. The collected RS images have high spatial resolutions, contain reservoirs with different spectral properties, contain urban areas as well as countrysides, and are acquired from different states and seasons. These factors increase the reliability and robustness of constructed models and the proposed workflow. The trained workflow was evaluated with an external testing dataset. Although the collected images are noisy in some areas and the RoIaR is in the countryside, the average $F_1$-scores of phase-1 and phase-2 outputs show the reliability of the prepared workflow. The workflow outperformed significantly in man-made object segmentation compared to the single-phase segmentation benchmark.

We suggest two relevant directions for future research: change detection and domain adaptation. An important possible application of RoIaR man-made object segmentation is the timely detection of unauthorized constructions around the reservoirs. This social problem might lead to serious consequences, such as reservoir contamination and dangerous situations for communities living in such places. Unfortunately, if such constructions are not detected in their first stages and local communities start to live there, it becomes more and more difficult for public services to move such communities. Hence, timely man-made object change detection in the RoIaR is an important application that might rely on the segmentation procedure described in this paper.

On the other hand, a key issue of RS imaging is the challenges in analyzing data from different locations and dates. Geographical and atmospheric variations affect the images, and domain adaptation approaches must often be developed. This problem has been circumvented in this paper by sparse annotation of all considered reservoirs, reflected by our sampling strategy. We are considering other possible domain adaptation approaches, such as few-shot and self-supervised learning. A context-aware network could be adopted as a possible alternative. This is left as future work since it involves its challenges.

## Code and Data Availability

Data supporting this study cannot be made available due to Google Earth's terms of service. The code is available through the GitHub repository [https://github.com/NayerehH/Man-made-objects-segmentation-in-RoI-around-reservoirs]

## Acknowledgments

The authors would like to thank FAPESP (Grant Nos. 2022/15304-4 and 2015/22308-2), CNPq, CAPES, FINEP, and MCTI PPI-SOFTEX (Grant No. TIC 13 DOU 01245.010222/2022-44).

## References

1. H. Gao, C. Birkett, and D. P. Lettenmaier, "Global monitoring of large reservoir storage from satellite remote sensing," *Water Resour. Res.* **48**(9) (2012).
2. L. Votruba and V. Broža, *Water Management in Reservoirs*, Elsevier (1989).
3. M. H. Almeer, "Vegetation extraction from free Google Earth images of deserts using a robust BPNN approach in HSV space," *Int. J. Adv. Res. Comput. Commun. Eng.* **2**(5) (2012).
4. Q. Hu et al., "Exploring the use of Google Earth imagery and object-based methods in land use/cover mapping," *Remote Sens.* **5**(11), 6026–6042 (2013).
5. T. Blaschke et al., "Geographic object-based image analysis—towards a new paradigm," *ISPRS J. Photogramm. Remote Sens.* **87**, 180–191 (2014).
6. N. Hamidishad and R. C. Junior, "Object-based method for identifying new constructions around water reservoirs: preliminary results," in *Anais Estendidos da XXXII Conf. Graphics, Patterns and Images*, SBC, pp. 172–175 (2019).
7. H. Ghanbari et al., "A meta-analysis of convolutional neural networks for remote sensing applications," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **14**, 3602–3613 (2021).
8. M. Wurm et al., "Deep learning-based generation of building stock data from remote sensing for urban heat demand modeling," *ISPRS Int. J. Geo-Inf.* **10**(1), 23 (2021).
9. B. Neupane, T. Horanont, and J. Aryal, "Deep learning-based semantic segmentation of urban features in satellite images: a review and meta-analysis," *Remote Sens.* **13**(4), 808 (2021).
10. B. Bischke et al., "Multi-task learning for segmentation of building footprints with deep neural networks," in *IEEE Int. Conf. Image Process. (ICIP)*, IEEE, pp. 1480–1484 (2019).
11. H. Zhao et al., "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2881–2890 (2017).
12. X. Yuan et al., "Land cover classification based on the PSPNet and superpixel segmentation methods with high spatial resolution multispectral remote sensing imagery," *J. Appl. Remote Sens.* **15**(3), 034511 (2021).
13. M. Li et al., "A deep learning method of water body extraction from high resolution remote sensing images with multisensors," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **14**, 3120–3132 (2021).
14. Y. Chen et al., "Extraction of urban water bodies from high-resolution remote-sensing imagery using deep learning," *Water* **10**(5), 585 (2018).
15. X. Zhang et al., "Land use mapping in the three gorges reservoir area based on semantic segmentation deep learning method," arXiv:1804.00498 (2018).
16. A. Van Soesbergen et al., "Dam reservoir extraction from remote sensing imagery using tailored metric learning strategies," *IEEE Trans. Geosci. Remote Sens.* **60**, 1–14 (2022).

17. K. Makantasis et al., "Deep learning-based man-made object detection from hyperspectral data," *Lect. Notes Comput. Sci.* **9474**, 717–727 (2015).
18. J. Yu et al., "A combined convolutional neural network for urban land-use classification with GIS data," *Remote Sens.* **14**(5), 1128 (2022).
19. E. Manos et al., "Convolutional neural networks for automated built infrastructure detection in the arctic using sub-meter spatial resolution satellite imagery," *Remote Sens.* **14**(11), 2719 (2022).
20. M. Khoshboresh-Masouleh, F. Alidoost, and H. Arefi, "Multiscale building segmentation based on deep learning for remote sensing RGB images from different sensors," *J. Appl. Remote Sens.* **14**(3), 034503 (2020).
21. M. Vakalopoulou et al., "Building detection in very high resolution multispectral data with deep learning features," in *IEEE Int. Geosci. and Remote Sens. Symp. (IGARSS)*, IEEE, pp. 1873–1876 (2015).
22. S. Ghaffarian and S. Ghaffarian, "Automatic building detection based on purposive fastica (PFICA) algorithm using monocular high resolution Google Earth images," *ISPRS J. Photogramm. Remote Sens.* **97**, 152–159 (2014).
23. B. Hou et al., "From W-net to CDGAN: bitemporal change detection via deep learning techniques," *IEEE Trans. Geosci. Remote Sens.* **58**(3), 1790–1802 (2019).
24. J. R. Jenson, *Introductory Digital Image Processing: A Remote Sensing Perspective*, Pearson (2015).
25. A. Jacobson et al., "A novel approach to mapping land conversion using Google Earth with an application to east Africa," *Environ. Modell. Software* **72**, 1–9 (2015).
26. J. Qian et al., "TCDNet: trilateral change detection network for Google Earth image," *Remote Sens.* **12**(17), 2669 (2020).
27. V. Visser et al., "Unlocking the potential of Google Earth as a tool in invasion science," *Biol. Invas.* **16**(3), 513–534 (2014).
28. U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," *Comput. Graphics Image Process.* **1**(3), 244–256 (1972).
29. M. Cubes, "A high resolution 3D surface construction algorithm," in *Proc. 14th Annu. Conf. Comput. Graphics and Interact. Tech.*, Association for Computing Machinery, New York, pp. 163–169 (1987).
30. L. da Fontoura Costa, *Shape Analysis and Classification: Theory and Practice*, CRC Press (2010).
31. O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," *Lect. Notes Comput. Sci.* **9351**, 234–241 (2015).
32. C. Ioffe and S. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," arXiv:1502.03167 (2015).
33. V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: a deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," arXiv:1505.07293 (2015).
34. L. Mou and X. X. Zhu, "RIFCN: recurrent network in fully convolutional network for semantic segmentation of high resolution remote sensing images," arXiv:1805.02091 (2018).
35. T.-Y. Lin et al., "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2117–2125 (2017).
36. A. Chaurasia and E. Culurciello, "LinkNet: exploiting encoder representations for efficient semantic segmentation," in *IEEE Vis. Commun. and Image Process. (VCIP)*, IEEE, pp. 1–4 (2017).
37. T.-Y. Lin et al., "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vision*, pp. 2980–2988 (2017).
38. C. H. Sudre et al., "Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations," *Lect. Notes Comput. Sci.* **10553**, 240–248 (2017).
39. S. Jadon, "A survey of loss functions for semantic segmentation," in *Proc. IEEE Conf. Comput. Intell. in Bioinf. and Comput. Biol. (CIBCB)* (2020).
40. D. Gupta, "Image segmentation keras: implementation of SegNet, FCN, Unet, PSPNet and other models in keras," arXiv:2307.13215 (2023).
41. P. Iakubovskii, "Segmentation models," 2019, https://github.com/qubvel/segmentation_models
42. D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv:1412.6980 (2014).

**Nayereh Hamidishad** received a PhD degree in computer science from the University of São Paulo (USP), Brazil, in 2023. She is interested in applying image processing, data mining, machine learning, and deep learning in varied fields, such as RS images, medical data, and energy systems.

**Roberto Marcondes Cesar Jr.** is a full professor in the Department of Computer Science at IME-USP. He served as the director of the eScience Research Center at USP and the head of the Computer Science Department. He was a member of the Image and Vision Computing and the Signal, Image, and Video Processing Editorial Boards, a chair and an invited speaker of conferences and workshops. His main research interests are in computer vision, machine learning, and artificial intelligence.