

A Semantic Segmentation Approach to River Sediment Identification

Ahmad Zaky Iswani¹, Kristiawan Dwi Usanto^{2*}

¹Department of Mechanical Engineering, College of Engineering, Southern Taiwan University of Science and Technology, Tainan 71005, TAIWAN

²Department of Electrical Engineering, Faculty of Intelligent Electrical and Informatics Technology, Institut Teknologi Sepuluh Nopember, Surabaya 60111, INDONESIA

Article Info

Abstract

Article history:

Received
04-08-2026

Revised
29-08-2026

Accepted
12-09-2026

Available Online
19-10-2026

Keywords:

Feature Pyramid
Network, LinkNet,
ResNet-50, River
Sediment Identification,
Semantic Segmentation,
U-Net

This study investigates the use of semantic segmentation to identify river sedimentation as part of soil erosion monitoring. Three neural network architectures, namely U-Net, LinkNet, and Feature Pyramid Network (FPN), were evaluated using 2,800 non-overlapping patches generated from 100 images of sediment formations from the IIUM River. After patch extraction, the image-mask pairs were divided into training and validation subsets using a 75:25 ratio, producing 2,100 training patches and 700 validation patches. Based on sediment-class Intersection over Union (IoU) under patch-level validation, U-Net achieved 0.8345, followed by LinkNet at 0.8189 and FPN at 0.2039. U-Net and LinkNet identified relatively uniform sediment formations more successfully, whereas all models struggled with irregular formations or sediment that was visually similar to river water. Because the data split was performed at the patch level, patches originating from the same source photograph may have been present in both the training and validation subsets. Therefore, the reported results should be interpreted as internal patch-level validation rather than as an estimate of generalisation to completely unseen source river images.



© 2026 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY 4.0) license <https://creativecommons.org/licenses/by/4.0/>.

Published by Universitas Katolik Indonesia Atma Jaya.

1. INTRODUCTION

Soil erosion has become an increasingly serious environmental issue due to rapid population growth and expanding human activities [1]. It can damage river ecosystems, reduce crop yields, lower water quality, and increase the risk of floods and landslides. At the IIUM Gombak Campus, sediment accumulation has affected the river by forming sand mounds and reducing water flow. These conditions may contribute to flooding during periods of heavy rainfall.

Computer vision has been widely used for image classification and object identification in fields such as buildings, vehicles, vegetation, and geographical monitoring. One important method is semantic segmentation, which assigns a specific class to every pixel in an image [2]. Unlike object detection, semantic segmentation produces detailed pixel-level predictions instead of using bounding boxes [3]. Therefore, it is more suitable for identifying sediment formations that may have irregular and unclear shapes.

Although semantic segmentation has been applied to tasks such as monitoring urban development and river water levels, its use for identifying river sedimentation remains limited. This study examines the feasibility of several semantic segmentation network architectures for river sediment identification. Since collecting and manually labelling a large dataset is difficult, the study trains different network models using a limited available dataset. Their performances are then compared to determine which architecture produces the most accurate sediment predictions.

2. LITERATURE REVIEW

Convolutional neural networks, commonly known as CNNs, have become important tools in computer vision because they can automatically learn relevant features from image data. Convolutional layers apply trainable kernels to detect visual patterns such as edges, shapes, and textures, while pooling

*Corresponding author : Kristiawan Dwi Usanto
Email address : 6022251013@student.its.ac.id

layers reduce feature-map dimensions and computational complexity. Nonlinear activation functions, particularly the Rectified Linear Unit, allow the network to represent complex relationships within high-dimensional data [4]. However, traditional CNNs are mainly designed for image classification and may lose spatial information when feature maps are flattened before being processed by fully connected layers.

Fully convolutional networks, or FCNs, were introduced to address the spatial limitations of conventional CNNs by replacing fully connected layers with convolutional operations [5]. FCNs generally consist of an encoder that extracts image features through convolution and downsampling, followed by a decoder that restores spatial dimensions through upsampling. Skip connections transfer information from encoder layers to corresponding decoder layers, allowing the model to recover spatial details that may be lost during downsampling. This encoder-decoder structure enables FCNs to perform pixel-level classification and makes them suitable for semantic segmentation.

U-Net is an FCN-based architecture originally proposed for biomedical image segmentation [6]. It has a symmetric U-shaped structure consisting of contracting encoder layers and expanding decoder layers connected through skip connections. The decoder concatenates upsampled feature maps with features from corresponding encoder layers, combining high-level contextual information with detailed spatial information. This design enables U-Net to generate accurate object boundaries, even when the available training dataset is relatively limited.

LinkNet is another encoder-decoder architecture developed to provide efficient semantic segmentation while reducing computational requirements [7]. Similar to U-Net, LinkNet connects encoder and decoder blocks at corresponding spatial levels through skip connections. However, LinkNet combines encoder and decoder features through summation rather than concatenation, which reduces the number of parameters and memory requirements. Its ability to process high-resolution images efficiently makes it suitable for environmental monitoring and other applications requiring relatively fast predictions.

Feature Pyramid Networks, or FPNs, use a top-down architecture with lateral connections to produce semantically meaningful feature maps at multiple spatial scales [8]. The encoder extracts hierarchical features, while the decoder and lateral connections combine high-level semantic information with detailed lower-level features. Feature maps from different levels are resized and combined before the final segmentation prediction is generated. This multiscale approach may be useful for sediment identification because river sediment formations can vary considerably in size, texture, shape, and spatial distribution.

Previous studies have also investigated sediment identification and transport using conventional machine learning techniques. Decision Trees, Random Forest, and Regularised Logistic Regression were used to classify sediment in a mixed-texture dredge pit based on bathymetric and side-scan sonar data. Random Forest achieved the highest average accuracy, although its performance decreased when the surface sediments consisted of mixed materials, such as mud and sand [9]. Gaussian Process Regression was also found to outperform Support Vector Machine in predicting sediment transport rates using hydraulic and sediment-related river characteristics [10].

Semantic segmentation has been widely applied to satellite and aerial imagery for environmental and geographical monitoring. EncNet, D-LinkNet, FCN, and LinkNet were compared using the Potsdam aerial image dataset, achieving overall accuracies ranging from 84.1% to 86.4% [11]. Several segmentation architectures were also evaluated for building extraction, with HRLinkNetV2 achieving the highest Intersection over Union score [12]. These findings indicate that architectural design and feature extraction strategies strongly influence segmentation performance in high-resolution geographical imagery.

Semantic segmentation has also been applied specifically to rivers and other environmental features. U-Net was used to distinguish rivers from land in synthetic aperture radar images, achieving mean Intersection over Union scores exceeding 0.95 [13]. DeepLabV3+ and SegNet were also compared for river-water segmentation, with DeepLabV3+ achieving the higher Intersection over Union score of 94.37% [14]. Semantic segmentation has additionally been successfully applied to urban development, road extraction, and power-line identification, demonstrating its suitability for analysing complex environmental imagery [15],[16].

One important challenge in semantic segmentation research is the requirement for large quantities of accurately labelled images. Environmental datasets are often difficult to collect because image acquisition depends on location, weather, accessibility, and suitable observation conditions. Manual pixel-level annotation is also time-consuming, particularly when objects have irregular or unclear boundaries. These limitations are especially relevant to river sedimentation because sediment formations may change continuously and may visually resemble water, mud, sand, or surrounding riverbed materials.

Data augmentation is frequently used to reduce overfitting when only a small dataset is available. Common augmentation methods include horizontal and vertical flipping, rotation, translation, cropping, and zooming, which increase the diversity of training samples without requiring additional field data.

Image flipping was reported to improve the mean Intersection over Union from 73.28% to 87.24%. Strong segmentation performance was also achieved by augmenting a limited medical-image dataset [17]. These findings suggest that appropriate data augmentation can improve model generalisation and support semantic segmentation under limited-data conditions.

Although previous research has demonstrated the effectiveness of semantic segmentation in medical imaging, urban monitoring, river extraction, and environmental surveillance, its direct application to river sediment formation remains limited. Sediment segmentation presents additional challenges because sediment boundaries may be unclear, formations may be irregular, and different sediment types may share similar visual characteristics. Furthermore, few studies have directly compared U-Net, LinkNet, and FPN using the same river-sediment dataset and training conditions. Therefore, this study evaluates these three architectures to determine which model provides the most accurate and reliable identification of river sediment formations under limited-data conditions.

3. METHODOLOGY

The experimental procedure consisted of four main stages, namely data collection, data preparation, parameter configuration, and model training. Three semantic segmentation architectures, specifically U-Net, LinkNet, and Feature Pyramid Network, were implemented using the segmentation_models library with a ResNet-50 backbone. The same experimental settings were applied to all architectures to ensure a fair and consistent comparison. Figure 1 presents the overall workflow employed in this study.

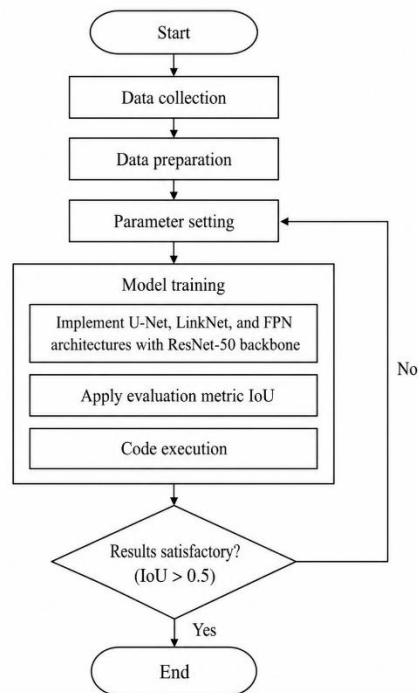


Figure 1. Experimental Setup Flowchart

Following data collection and preparation, the training parameters were configured before the three models were implemented and trained. Their segmentation performances were evaluated using the Intersection over Union metric. When the results did not meet the predefined criterion of an IoU score greater than 0.5, the parameter configuration and training process were repeated. Models that achieved satisfactory results were retained for further quantitative and qualitative analysis.

3.1 Data Collection

A total of 100 images were collected from different locations along the IIUM River to capture variations in river conditions, sediment formations, and surrounding backgrounds. The images were acquired using a Sony Xperia 1 smartphone equipped with a 12-megapixel main camera. Each raw image had a resolution of 4032×2268 pixels and was stored in PNG format. This collection strategy was intended to provide representative visual information for training and evaluating the semantic segmentation models. Figure 2 shows an example image and its corresponding ground-truth mask.



Figure 2. Example of a river image and its corresponding ground-truth segmentation mask

Each image was manually annotated using the APEER annotation platform to generate a corresponding ground-truth mask. The masks contained four classes, namely unlabeled regions, river water, river sediment, and background, represented by grayscale values of 0, 1, 2, and 3, respectively. Although the unlabeled class was not relevant to the sediment identification task, it was retained because it was automatically included by the annotation tool.

3.2 Dataset Preparation

The original dataset consisted of 100 river image-mask pairs. Each image and mask was first adjusted to dimensions divisible by the selected patch size and then divided into non-overlapping 512 x 512 pixel patches using a step size of 512 pixels. This procedure produced a total of 2,800 image-mask patch pairs. The training and validation split was performed after patch extraction, at the patch level, using a 75:25 ratio. The width and height of each 4032 x 2268 pixel image were cropped to 3584 x 2048 pixels, the nearest dimensions divisible by the selected patch size, while preserving pixel-level alignment with the corresponding mask. Each image-mask pair therefore produced 28 patches (7 patches across the width and 4 patches across the height). The 512 x 512 patch size was selected as a compromise between retaining sufficient contextual information and meeting the memory limitations of the available training computer.

The resulting 2,800 patch pairs were then divided using the splitfolders library into 2,100 training patches and 700 validation patches. Because the split was performed after patch extraction, patches originating from the same source photograph may have been present in both subsets. This represents a methodological limitation because the validation subset was not fully independent at the source-image level and may yield an optimistic estimate of performance. Accordingly, the results of this study are interpreted as internal patch-level validation rather than as a measure of generalisation to completely unseen source river photographs. Future studies should perform the split at the source-image level before patch extraction.

3.2.1 Parameter Setting

Following dataset preparation, the training parameters were defined to ensure that all segmentation models were evaluated under consistent conditions. The batch size was set to eight, which was the maximum value supported by the available computational memory. A larger batch size could provide more stable gradient estimates, but it would also increase memory consumption during training. Therefore, the same batch size was applied to U-Net, LinkNet, and FPN. The preprocessing and augmentation settings are summarised in Table 1.

The number of output classes was set to four based on the grayscale values contained in the ground-truth masks. These classes represented unlabeled regions, river water, river sediment, and background, with class identifiers of 0, 1, 2, and 3, respectively. Although the main segmentation task focused on river

water, sediment, and background, the unlabeled class was retained because it was automatically generated by the annotation platform and could not be removed from the masks. The grayscale class labels were subsequently converted into categorical representations using one-hot encoding.

A ResNet-50 network pretrained on ImageNet was used as the encoder backbone for all three architectures. Each input image was preprocessed using the preprocessing function associated with the ResNet-50 backbone before being passed to the model. The use of the same backbone provided a common feature-extraction structure and allowed the performance differences to be attributed primarily to the decoder designs of U-Net, LinkNet, and FPN. The masks were reshaped and converted using the Keras `to_categorical` function to produce four-channel target outputs.

Table 1. Parameter settings used for dataset preprocessing and augmentation

Parameter	Value
Batch size	8
Number of classes	4
Encoder backbone	ResNet-50
Pretrained weights	ImageNet
Mask encoding	One-hot categorical encoding
Image augmentation	Horizontal and vertical flipping
Image generator	Keras ImageDataGenerator
Image and mask seed	Identical seed value

Data augmentation was applied to increase the diversity of the limited training dataset. Horizontal and vertical flipping were performed using the Keras ImageDataGenerator, with identical transformations applied to the image patches and their corresponding masks. The image and mask generators used the same seed value to maintain correct spatial alignment between each augmented image and its annotation. Rotation and scaling were not applied since interpolation during these transformations could modify the discrete pixel values of the segmentation masks.

3.2.2 Model Training

Model training was conducted after completing data collection, dataset preparation, and parameter configuration. Three semantic segmentation architectures, namely U-Net, LinkNet, and Feature Pyramid Network, were implemented using TensorFlow Keras and the `segmentation_models` library. Each architecture used an ImageNet-pretrained ResNet-50 backbone, an input size of 512×512 pixels, and four output classes. The models were trained using the same dataset, hyperparameters, loss function, and evaluation metric to provide a consistent comparison.

3.3 Network Architecture Implementation

The implemented U-Net, LinkNet, and FPN architectures differed primarily from their original forms through the inclusion of a pretrained backbone in the encoder phase. The same ResNet-50 backbone was applied to all three models to provide a common feature-extraction structure. Therefore, the main architectural differences were located in the decoder phase, particularly in the upsampling methods and skip-connection operations. These differences were examined to determine how each decoder design affected river-sediment segmentation.

A backbone is a pretrained network used in the encoder phase to extract relevant features from an input image. Networks trained on large image datasets contain previously learned weights that can detect general visual features such as lines, curves, edges, textures, and shapes. Integrating these weights into a semantic segmentation architecture can reduce the amount of training required and improve feature extraction when the available dataset is limited. In this study, the same pretrained backbone was used in U-Net, LinkNet, and FPN.

ResNet-50 was selected as the backbone because it provides a balance between network depth, feature-extraction capability, and computational requirements. The architecture consists of residual blocks containing convolutional layers and identity connections. Within a residual block, an input x is processed through weighted layers to obtain a residual function $F(x)$, which is then added to the original input. The output of the residual block is expressed as:

$$H(x) = F(x) + x \quad (1)$$

The identity connection allows information to pass directly across the residual block and supports more efficient learning in deep networks. Since ResNet-50 is positioned in the encoder phase, its residual stages generate feature maps at different spatial scales and levels of abstraction. Lower-level stages

capture features such as edges and textures, while deeper stages extract more complex visual patterns. Figure 3 illustrates the residual block used in the ResNet architecture.

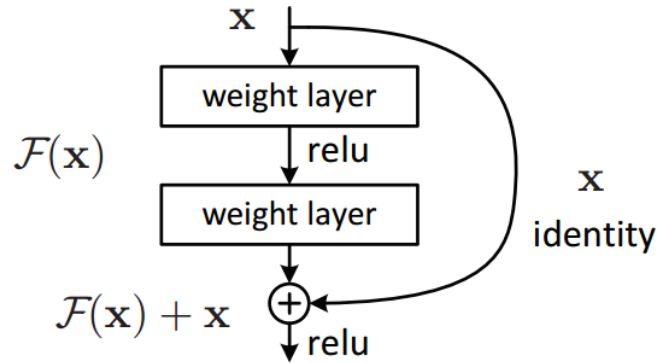


Figure 3. Residual block used in the ResNet architecture

The ResNet-50 backbone begins with a 7×7 convolutional layer containing 64 filters and a stride of two. This layer is followed by a 3×3 max-pooling operation that further reduces the spatial dimensions of the input feature map. The remaining network is divided into four residual stages containing combinations of 1×1 , 3×3 , and 1×1 convolutional layers. The arrangement of the ResNet-50 stages used in this study is shown in Figure 4 and summarised in Table 2.

Table 2. Internal structure of the ResNet-50 backbone

Layer stage	Convolutional configuration	Repetition
Conv1	7×7 convolution, 64 filters, stride 2	1
Conv2_x	3×3 max pooling, stride 2; 1×1 , 64; 3×3 , 64; 1×1 , 256	3
Conv3_x	1×1 , 128; 3×3 , 128; 1×1 , 512	4
Conv4_x	1×1 , 256; 3×3 , 256; 1×1 , 1024	6
Conv5_x	1×1 , 512; 3×3 , 512; 1×1 , 2048	3

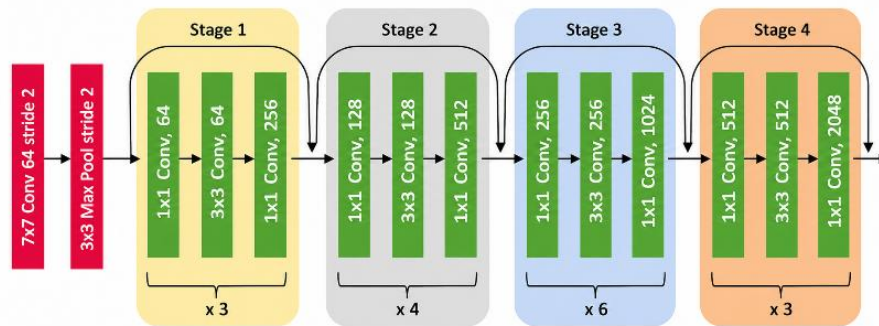


Figure 4. Schematic structure of the ResNet-50 backbone

The implemented U-Net used the ResNet-50 stages as its encoder and transpose convolution as its upsampling method. After the final ResNet stage, the encoded tensor was upsampled using a 4×4 transpose convolution with a stride of two. The first decoder feature map was concatenated with the output of the corresponding ResNet stage, allowing contextual information from the deeper layers to be combined with spatial information from the encoder. Two 3×3 convolutional operations with Rectified Linear Unit activation were then applied before the tensor was upsampled to the next decoder level.

The decoder progressively generated feature maps containing 128, 64, 32, and 16 channels. At each level, the decoder features were concatenated with features from the corresponding encoder stage. The final decoder output was processed using a 1×1 convolution followed by softmax activation. This produced an output segmentation map with dimensions of 512×512 pixels and four class-probability channels. Figure 5 shows the implemented U-Net architecture.

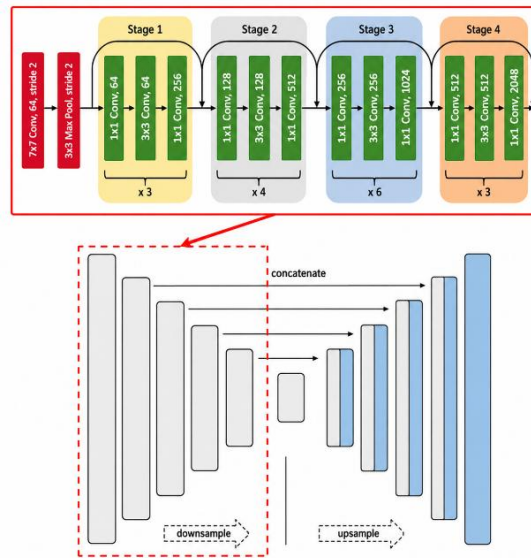


Figure 5. Implemented U-Net architecture with the ResNet-50 backbone

The implemented LinkNet followed an encoder-decoder structure similar to U-Net and also used transpose convolution for upsampling. However, its skip connections combined encoder and decoder feature maps through element-wise summation rather than concatenation. After the final ResNet stage, the feature tensor was upsampled using a 4×4 transpose convolution with a stride of two. The resulting feature map was then added to the corresponding ResNet feature map.

Each decoder level applied two 3×3 convolutional layers with Rectified Linear Unit activation before proceeding to the next upsampling stage. The decoder generated feature maps containing 128, 64, 32, and 16 channels. A final 1×1 convolution and softmax activation produced a four-channel output with dimensions of 512×512 pixels. The implemented LinkNet architecture is shown in Figure 6.

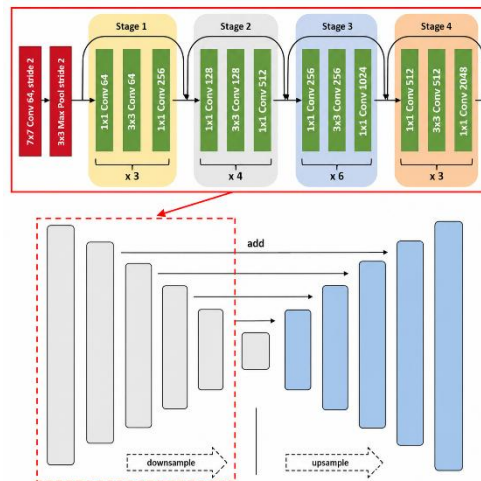


Figure 6. Implemented LinkNet architecture with the ResNet-50 backbone

The implemented FPN differed from U-Net and LinkNet by constructing a feature pyramid from several ResNet-50 stages. Nearest-neighbour interpolation was used instead of transpose convolution to increase the spatial dimensions of the feature maps. Features obtained from different encoder stages were transferred through lateral connections and combined through element-wise summation. A 1×1 convolution was applied before each summation operation to provide compatible channel dimensions.

Each pyramid level was subsequently processed using two consecutive 3×3 convolutional operations. The feature maps were then upsampled by different factors until they reached identical spatial dimensions. After resizing, the pyramid features were concatenated to form a multiscale feature representation. A final 3×3 convolution followed by softmax activation generated the four-class segmentation output. Figure 7 shows the implemented FPN architecture.

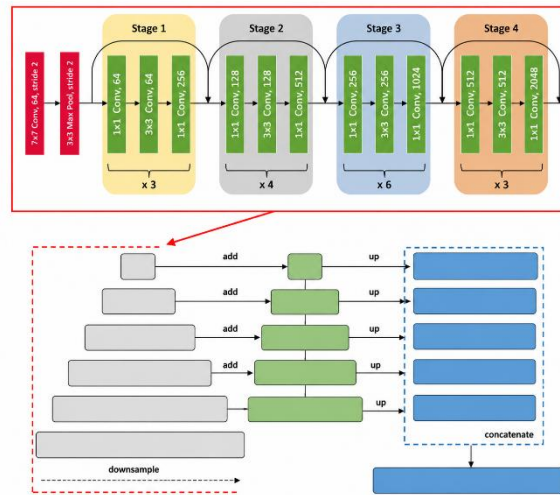


Figure 7. Implemented FPN architecture with the ResNet-50 backbone

Comparison of implemented architectures. Although the three architectures used the same ResNet-50 encoder, they employed different decoder structures and feature-combination operations. U-Net concatenated corresponding encoder and decoder features, while LinkNet combined them through summation. FPN constructed multiple pyramid levels and combined multiscale features using summation preceded by 1×1 convolution. The principal characteristics of the implemented architectures are summarised in Table 3.

Table 3. Comparison of the implemented segmentation architectures

Characteristic	U-Net	LinkNet	FPN
General shape	Generally U-shaped	Generally U-shaped	Pyramidal, multicolumn
Backbone	ResNet-50	ResNet-50	ResNet-50
Pretrained weights	ImageNet	ImageNet	ImageNet
Upsampling method	Transpose convolution	Transpose convolution	Nearest-neighbour interpolation
Skip-connection operation	Concatenation	Summation	Summation preceded by 1×1 convolution
Feature representation	Encoder-decoder features	Encoder-decoder features	Multiscale pyramid features
Output activation	Softmax	Softmax	Softmax
Output dimensions	$512 \times 512 \times 4$	$512 \times 512 \times 4$	$512 \times 512 \times 4$

3.4 Evaluation Metric and Loss Function

Intersection over Union was selected as the principal evaluation metric for measuring segmentation performance. IoU compares the overlapping area between the predicted segmentation region and the corresponding ground-truth region with their total combined area. A higher IoU score indicates greater agreement between the predicted and manually annotated masks. This metric is commonly used for semantic segmentation because it directly evaluates the quality of the predicted object region.

IoU was preferred over conventional pixel accuracy due to the possibility of class imbalance. In a dataset containing a large background region, accuracy may appear high even when the model performs poorly on smaller classes such as river sediment. IoU does not depend on the number of correctly identified true-negative pixels and therefore provides a more representative evaluation of the target classes. The IoU score was calculated as:

$$IoU = \frac{TP}{TP+FP+FN} \quad (2)$$

Where TP represents correctly predicted target-class pixels, FP represents pixels incorrectly assigned to the target class, and FN represents target-class pixels that were not correctly identified. Class-specific IoU was calculated separately for river water, river sediment, and background. The automatically generated unlabeled class was retained in the four-channel model output but was excluded from the final

reported mean IoU because it was not a meaningful target class for this study. Accordingly, the final mean IoU was calculated as the arithmetic mean of the river-water, river-sediment, and background IoU values.

Jaccard loss was used as the loss function during model training. IoU and the Jaccard index describe the same overlap measurement, although the term IoU is generally used for the evaluation metric while Jaccard is used to describe the training loss. The loss function guided the optimization process by penalising differences between the predicted and ground-truth masks. River-sediment IoU was treated as the main class-specific indicator because sediment identification was the principal objective of this study.

3.5 Training and Evaluation Procedure

The numbers of training and validation steps per epoch were determined by dividing the number of image-mask pairs by the batch size. With 2,100 training samples and 700 validation samples, the models were trained using a batch size of eight for 100 epochs. U-Net, LinkNet, and FPN were compiled separately using the Adam optimizer, Jaccard loss, and Intersection over Union as the evaluation metric. All architectures used the same ResNet-50 backbone, ImageNet-pretrained weights, four output classes, and softmax activation.

Validation performance was monitored after each training epoch using the IoU metric configured during model training, and these epoch-wise values were used to describe convergence behaviour. After training, final model performance was evaluated using class-specific IoU values on the validation patches. The final mean IoU reported in Table 5 was calculated as the arithmetic mean of the river-water, river-sediment, and background IoU values. The automatically generated unlabeled class was excluded because it was not a meaningful target class for the study. The epoch-wise IoU curves and the final class-specific IoU values in Table 5 therefore serve different reporting purposes. All reported evaluation results were obtained from the patch-level split.

Table 4. Computer specifications used for model training

Component	Specification
Computer model	Alienware m15 Ryzen Edition R5
Processor	AMD Ryzen 9 5900HX at 3.30 GHz
System memory	16 GB RAM
Graphics processor	NVIDIA GeForce RTX 3070 Laptop GPU
Dedicated graphics memory	8 GB

All experiments were conducted using the same computational platform to maintain consistent training conditions. The models were trained on an Alienware m15 Ryzen Edition R5 laptop equipped with an AMD Ryzen 9 5900HX processor, 16 GB of RAM, and an NVIDIA GeForce RTX 3070 Laptop GPU with 8 GB of dedicated memory. The available GPU memory limited the maximum batch size to eight. Table 4 summarises the computer specifications.

4. RESULT AND DISCUSSION

The performances of U-Net, LinkNet, and Feature Pyramid Network were evaluated through quantitative and qualitative analyses. The quantitative analysis examined the progression of training and validation losses, training and validation Intersection over Union scores, and the class-specific IoU scores obtained after training. The qualitative analysis compared the predicted segmentation masks with the corresponding ground-truth masks for selected validation patches. Because the data split was performed at the patch level, these evaluations are used to compare model performance under an internal patch-level validation setting and are not used to claim generalisation to fully independent source images.

4.1 Training and Validation Performance

The training and validation losses and IoU scores were recorded throughout 100 training epochs. These curves were used to examine how each model learned from the training dataset and how effectively it performed on the validation dataset. In general, all three architectures showed rapid improvement during the early epochs, followed by slower convergence during the remaining training period. However, the validation curves were considerably more irregular than the training curves.

For U-Net, the training loss began at approximately 0.66 and decreased sharply to around 0.47 after the first 10 epochs. It subsequently declined more gradually and reached a value slightly below 0.40 at the end of 100 epochs. The validation loss initially fluctuated between approximately 1.05 and 1.15 before rapidly decreasing to around 0.50 within the first five epochs. Although the validation loss remained irregular until approximately epoch 22, it eventually stabilised slightly above 0.40. The U-Net training curves are shown in Figure 8.

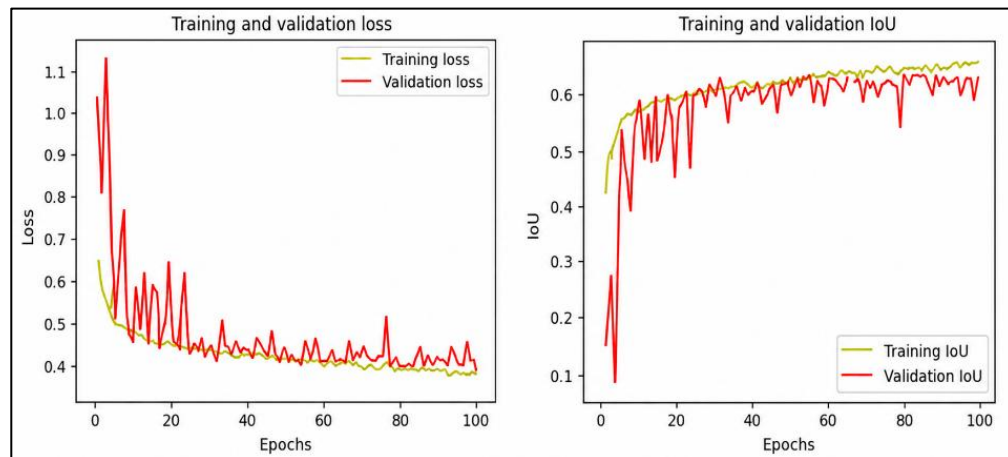


Figure 8. U-Net loss and IoU across training epochs

The U-Net training IoU increased rapidly from approximately 0.40 to 0.57 during the first 10 epochs. After this initial improvement, the score increased gradually and reached approximately 0.63 at the end of training. The validation IoU increased sharply to approximately 0.53 during the first five epochs but remained irregular until approximately epoch 30. After 100 epochs, the validation IoU reached approximately 0.60.

The LinkNet training loss started at approximately 0.75 and decreased to around 0.50 during the first 15 epochs. It continued to decline gradually and reached a value slightly above 0.40 after 100 epochs. The validation loss also decreased rapidly during the initial training period but showed considerable fluctuations before stabilising. These variations were more pronounced than those observed in the corresponding training loss. The LinkNet training curves are shown in Figure 9.

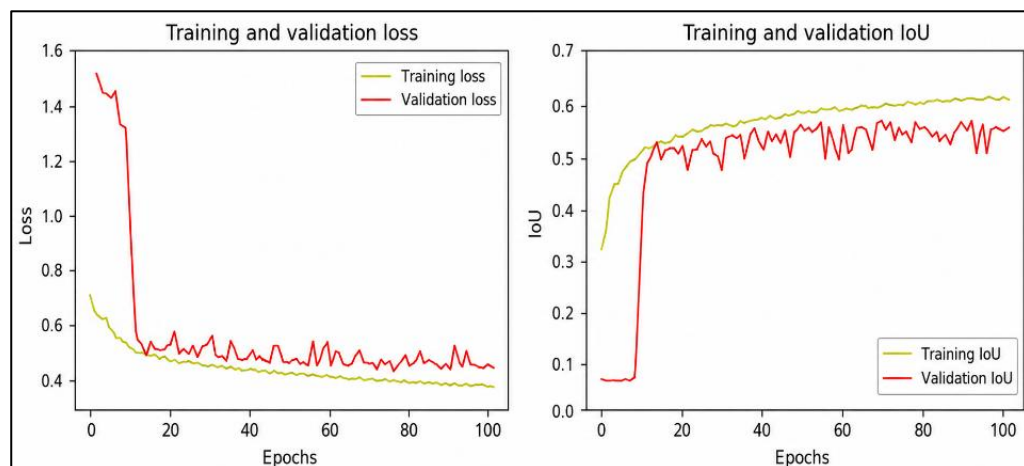


Figure 9. LinkNet loss and IoU across training epochs

The LinkNet training IoU increased rapidly during the early epochs and then improved more gradually as training progressed. The validation IoU rose from approximately 0.50 to 0.57 during the initial epochs. It remained irregular for approximately the next 20 epochs, including a temporary decline to around 0.42 near epoch 18. The score later stabilised and reached approximately 0.60 at the end of training.

The FPN training and validation curves showed a similar general pattern of rapid change during the early epochs followed by slower convergence. However, similarity in the aggregate epoch-wise curves does not indicate that FPN achieved sediment-segmentation performance comparable to U-Net and LinkNet. The class-wise evaluation revealed a clear weakness in the sediment class: FPN obtained a sediment IoU of 0.2039, substantially below U-Net at 0.8345 and LinkNet at 0.8189. Thus, the aggregate curves in Figure 10 should be interpreted together with class-specific evaluation.

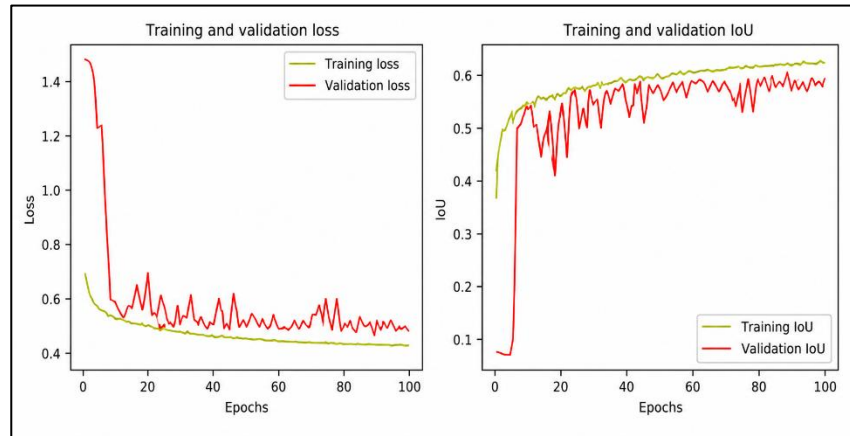


Figure 10. FPN loss and IoU across training epochs

A common characteristic across the three architectures was the jagged behaviour of the validation curves. Although the fluctuations became less pronounced during later epochs, they remained visible throughout the training process. In contrast, the training curves showed a smoother and more consistent progression. This difference suggests that the trained models may have experienced overfitting or may have been sensitive to variations within the relatively limited validation dataset.

Although the aggregate validation IoU curves occupied broadly similar ranges during training, the final class-specific evaluation showed substantial differences among the architectures. FPN achieved a river-water IoU of 0.9217 and a background IoU of 0.9967, but its sediment IoU was only 0.2039. In contrast, U-Net and LinkNet achieved sediment IoU values of 0.8345 and 0.8189, respectively. These results demonstrate that aggregate validation behaviour alone is insufficient for evaluating sediment-identification performance and should be interpreted together with class-specific IoU.

4.2 Class-Wise IoU Performance

The trained models were evaluated using mean IoU and individual IoU scores for the river-water, sediment, and background classes on the 700 validation patches. The unlabeled class was included during model training but was not considered a meaningful target class for the interpretation of sediment-identification performance. Since the main objective of this study was to identify river sediment, sediment-class IoU was treated as the principal class-specific performance indicator. All values in this section represent internal patch-level validation results.

Table 5. Mean and class-specific IoU scores of the implemented architectures

IoU class	U-Net	LinkNet	FPN
Mean IoU	0.9162	0.8745	0.7074
River	0.9783	0.8949	0.9217
Sediment	0.8345	0.8189	0.2039
Background	0.9360	0.9098	0.9967

U-Net achieved the highest mean IoU of 0.9162, followed by LinkNet at 0.8745 and FPN at 0.7074. U-Net also achieved the highest river-water IoU of 0.9783. Although FPN achieved a river-water IoU of 0.9217, its sediment-class performance was substantially lower than those of the other architectures. The high background IoU obtained by FPN indicates that it was effective at identifying background pixels but frequently failed to distinguish sediment from the other classes.

For the sediment class, U-Net achieved the highest IoU score of 0.8345. LinkNet produced a slightly lower score of 0.8189, indicating that both architectures were relatively effective at identifying sediment formations within the patch-level validation setting. In contrast, FPN achieved a sediment IoU of only 0.2039. Based on these internal validation results, U-Net provided the strongest quantitative performance, followed by LinkNet and FPN.

4.3 Qualitative Segmentation Comparison

A qualitative analysis was conducted by comparing the predicted masks with the ground-truth masks of selected validation patches. The samples represented different river conditions, viewing angles, backgrounds, and sediment formations. The comparison focused on whether each model correctly identified river water, river sediment, and background regions. Particular attention was given to the models' ability to segment uniform and irregular sediment formations.

Figure 11 presents a river image containing relatively uniform sediment formations on both sides of the riverbank. U-Net correctly identified most of the sediment pixels and produced a prediction that

closely followed the ground-truth mask. LinkNet also identified the main sediment regions but incorrectly assigned some pixels on the upper bank to the river-water class. FPN produced more extensive misclassification on both sides of the riverbank.

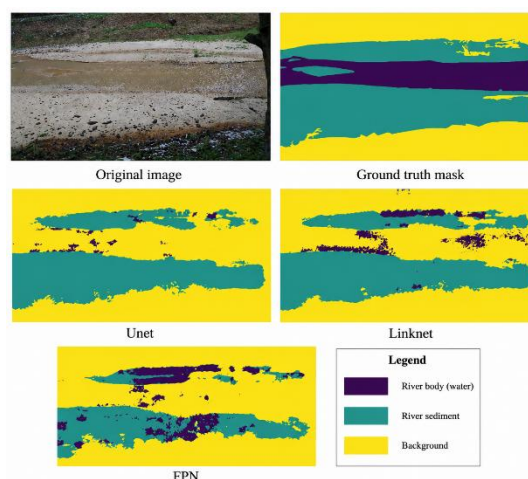


Figure 11. Original image, ground-truth mask, and predicted masks produced by U-Net, LinkNet, and FPN for sample image 1

The results for sample image 1 correspond to the class-specific IoU scores. U-Net generated the most accurate sediment prediction, while LinkNet produced a similar result with more mislabelled pixels. FPN had greater difficulty distinguishing sediment from river water and background. The relatively uniform shape and colour of the sediment enabled U-Net and LinkNet to identify the main formations more successfully.

Figure 12 presents another image containing a relatively uniform sediment formation. All three models incorrectly predicted the presence of a sediment formation on the right bank, although this region was not labelled as sediment in the ground-truth mask. U-Net nevertheless identified most of the true sediment pixels on the left bank. LinkNet misclassified part of the sediment as background, while FPN identified much of the true sediment region as river water.

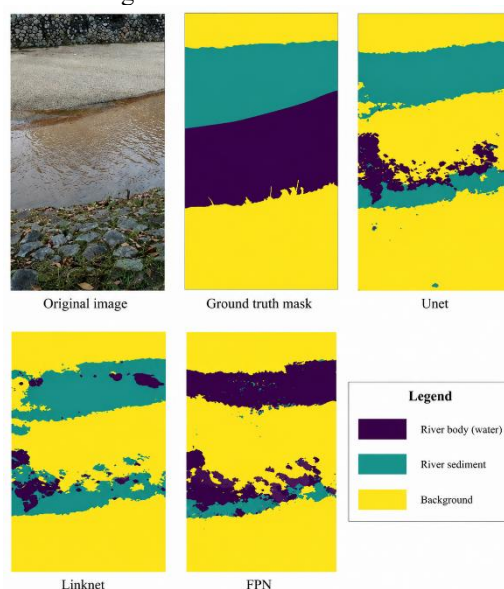


Figure 12. Original image, ground-truth mask, and predicted masks produced by U-Net, LinkNet, and FPN for sample image 2

The predictions for sample image 2 again showed that U-Net provided the closest agreement with the ground truth. LinkNet identified the general sediment location but produced more fragmented and incorrect classifications. FPN generated substantial confusion between the river-water and sediment classes. These findings indicate that U-Net and LinkNet were more capable of identifying sediment when the formation was visually continuous and relatively uniform.

Figure 13 represents a more difficult segmentation case in which the colour and texture differences among river water, sediment, and background were less distinct. U-Net correctly identified part of the river-water region but incorrectly classified the sediment as river water. LinkNet and FPN produced less accurate results and assigned most of the image to the background class. The reduced visual distinction between classes therefore affected all three models.

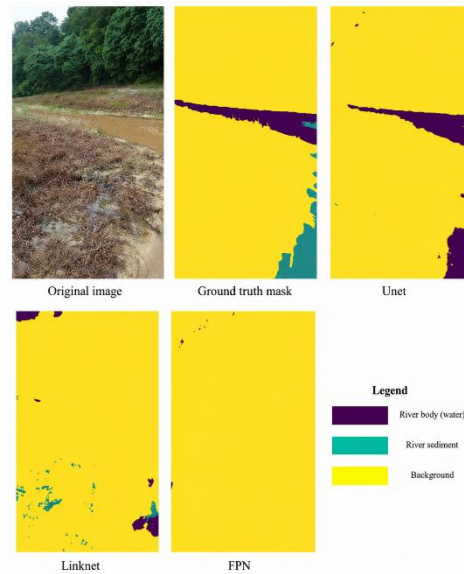


Figure 13. Original image, ground-truth mask, and predicted masks produced by U-Net, LinkNet, and FPN for sample image 3

Figure 14 presents an image containing isolated sediment clusters within the river rather than a single continuous sediment formation. The colour and texture differences between river water and sediment were also relatively small. None of the three models successfully identified the isolated sediment clusters. The predicted masks primarily classified the area as river water or background.

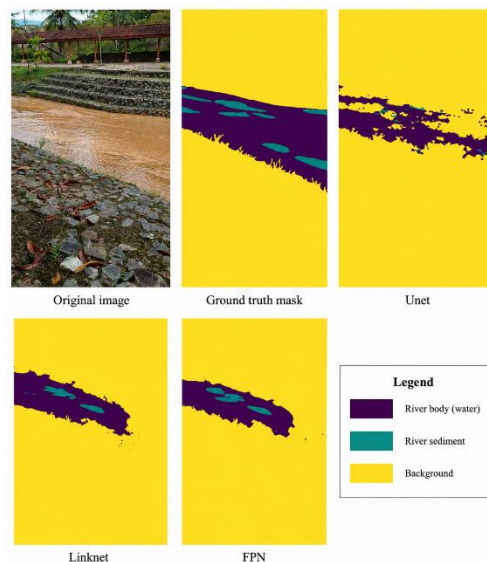


Figure 14. Original image, ground-truth mask, and predicted masks produced by U-Net, LinkNet, and FPN for sample image 4

The qualitative results demonstrate that the architectures performed differently depending on the shape and visual characteristics of the sediment. When sediment formations were uniform and clearly distinguishable from their surroundings, U-Net produced the most accurate masks, followed by LinkNet. When the formations were irregular, isolated, or visually similar to river water, the performance of all models declined. These observations show that high sediment-class IoU under patch-level validation did not guarantee reliable identification across all sediment conditions represented in the validation patches.

4.4 Discussion

The quantitative and qualitative results under the internal patch-level validation setting indicate that U-Net outperformed LinkNet and FPN in river-sediment identification. U-Net achieved the highest mean IoU and sediment-class IoU while also producing fewer visible misclassifications in the selected validation patches. LinkNet achieved a sediment IoU close to that of U-Net but generated slightly more classification errors. FPN obtained the lowest sediment IoU and showed considerably more confusion between sediment, river water, and background. These comparisons should be interpreted within the limitations of the patch-level data split.

One possible explanation for the difference between U-Net and LinkNet is the method used to combine encoder and decoder features. U-Net uses concatenation, while LinkNet uses element-wise summation. In LinkNet, corresponding feature maps are added together to form a single output feature map. In U-Net, encoder features are appended to decoder features, allowing both sets of information to remain available for subsequent convolutional processing.

The concatenation operation may preserve more detailed spatial information from the encoder. This may allow the U-Net decoder to use both high-level contextual information and lower-level boundary details more effectively. In contrast, summation blends the encoder and decoder feature maps directly, which may reduce the distinction between some feature representations. This difference may have contributed to the slightly better segmentation performance obtained by U-Net.

FPN produced substantially weaker sediment predictions than U-Net and LinkNet despite its apparently similar aggregate validation-curve progression. Its decoder structure differs from those of U-Net and LinkNet because it constructs several pyramid levels and processes features through multiple lateral and convolutional operations. The results therefore demonstrate the importance of separating convergence behaviour from class-specific segmentation quality: stable or comparable aggregate curves do not imply that all classes are segmented equally well.

The class-wise results showed that FPN achieved strong performance for river water and background but failed to identify sediment reliably. Specifically, FPN achieved IoU values of 0.9217 for river water and 0.9967 for background, compared with only 0.2039 for sediment. This imbalance explains why an aggregate validation metric may appear acceptable even when the target sediment class performs poorly. One possible explanation is the small colour and texture difference between sediment and river water, which may make subtle sediment features difficult to preserve during repeated multiscale feature processing.

The irregular validation curves suggest that the models may have experienced overfitting or sensitivity to variations within the relatively limited validation patch set. The qualitative analysis supports this interpretation because the architectures performed relatively well for uniform sediment formations but failed when sediment appeared as isolated or irregular clusters. However, because the training and validation subsets were created at the patch level, these observations should not be interpreted as evidence of model generalisation to completely unseen source photographs.

Another factor that may explain the segmentation errors is the large intraclass variation within the background class. The background label included visually different objects such as stone walls, vegetation, soil, riverbanks, and other surrounding structures. These objects share the same class label despite having different colours, textures, and shapes. Such variation can make it difficult for the models to learn a consistent representation of the background class.

At the same time, the river-water and sediment classes sometimes exhibited small interclass differences. In several images, sediment and river water had similar colours and textures. The models could therefore confuse pixels belonging to these two classes, particularly when sediment formations were thin, isolated, or partially submerged. The combination of large intraclass differences and small interclass differences likely contributed to the observed classification errors.

A possible improvement is to divide the broad background category into several more specific classes. For example, stone walls, vegetation, riverbanks, and exposed soil could be assigned separate labels. Increasing the number of classes may allow the models to learn more consistent visual characteristics for each object type. This approach may also reduce confusion between background features and sediment formations.

Further improvement may be achieved by using a larger and more diverse dataset. Additional images should include different lighting conditions, river-water colours, sediment textures, observation angles, and irregular sediment formations. More detailed annotation may also be required for boundaries where river water and sediment overlap. Data augmentation methods beyond flipping could be investigated, provided that the discrete mask labels are preserved.

Newer semantic segmentation architectures may also be evaluated. The `segmentation_models` library provides convenient implementation of the architectures used in this study, but it limits the available models to those included in the library. More recent architectures may offer improved feature

preservation, attention mechanisms, or multiscale context processing. However, implementing such models may require additional programming, computational resources, and consideration of software licensing conditions.

Table 6 summarises the performance of the three architectures under the patch-level validation setting. U-Net provided the strongest sediment identification with fewer visible classification errors, LinkNet achieved comparable sediment IoU with slightly more mislabelled regions, and FPN showed substantially weaker sediment detection and the greatest number of visible errors.

Table 6. Summary of the performance of the implemented architectures

Performance aspect	U-Net	LinkNet	FPN
Epoch progression	Initial fluctuations followed by gradual stabilisation	Initial fluctuations followed by gradual stabilisation	Initial fluctuations followed by gradual stabilisation
Sediment IoU	0.8345	0.8189	0.2039
Uniform sediment prediction	Good, with fewer mislabelled pixels	Good, with slightly more mislabelled pixels	Moderate, with considerably more mislabelled pixels
Irregular sediment prediction	Poor	Poor	Poor
Overall performance	Best	Second best	Lowest

Overall, U-Net was the most effective architecture under the patch-level validation setting used in this study. Its concatenation-based skip connections may have enabled better preservation of encoder information and more accurate reconstruction of sediment boundaries. LinkNet remained a competitive alternative but produced slightly more classification errors. FPN was less effective for sediment identification, particularly when sediment formations were irregular or visually similar to river water. Because the data were split at the patch level, this ranking should not be interpreted as evidence of performance on completely unseen source photographs.

5. CONCLUSION

This study investigated the feasibility of semantic segmentation for identifying river sediment formations in the IIUM River using U-Net, LinkNet, and Feature Pyramid Network with a common ResNet-50 backbone. Under the internal patch-level validation setting, U-Net achieved the highest sediment-class IoU at 0.8345, followed by LinkNet at 0.8189 and FPN at 0.2039. These results demonstrate the feasibility of applying semantic segmentation to river-sediment identification under the evaluation setting used in this study. However, because the training and validation subsets were separated after patch extraction, the reported results should not be interpreted as evidence of generalisation to completely unseen source river photographs.

The qualitative evaluation showed that U-Net and LinkNet were generally able to identify sediment when the formations were uniform and visually distinguishable from the surrounding river water and background. However, all three architectures performed poorly when sediment formations were irregular, isolated, or visually similar to the river body. These findings describe behaviour within the patch-level validation set and should be interpreted cautiously because patches originating from the same source photograph may have appeared in both training and validation subsets.

Future studies should improve the class-labelling strategy by dividing the broad background category into more specific object classes, such as vegetation, stone walls, exposed soil, and riverbanks. A larger and more diverse dataset should also be collected, and the training and validation datasets should be separated at the source-image level before patch extraction to provide an independent assessment of generalisation. More robust segmentation architectures, particularly those incorporating dilated or atrous convolution, may also be investigated to better capture features at different spatial scales.

AUTHOR CONTRIBUTIONS

Ahmad Zaky Iswani: conceptualization, methodology, investigation, data curation, software, formal analysis, visualization, writing (original draft). Kristiawan Dwi Usmanto: methodology, validation, supervision, writing (review and editing). Both authors have read and approved the final manuscript.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

ETHICAL CLEARANCE STATEMENT

This study used photographs of the river environment that did not contain identifiable people or personal data. The research did not involve human participants, and human-subject ethical clearance was not required.

DECLARATION OF AI TOOL USAGE

During the preparation of this manuscript, the authors used ChatGPT (OpenAI) to assist with English-language editing and checking the consistency of figure and table references. The authors reviewed and edited the AI-assisted output and take full responsibility for the final content.

REFERENCE

- [1] X. Ren, C. Huang, Z. Zhu, L. Wang, N. Wang, X. Mu, D. Luo, J. Zhang, L. Xiang and X. Huang, "Anthropogenic forest degradation and its impact on soil erosion in the mountainous region of northern China," *Catena*, pp. 1-11, 2024.
- [2] J. M. Molina, J. P. Llerena, L. Usero and M. A. Patricio, "Advances in instance segmentation: Technologies, metrics and applications in computer vision," *Neurocomputing*, vol. 625, pp. 1-19, 2025.
- [3] M. Hasan, A. Khatun, S. A. Juie, S. B. Shorif and M. S. Uddin, "Deep Learning Based Semantic Image Segmentation: A Problem Driven Analysis of Architectures, Datasets, and Open Challenges," *Journal of Computer and Communications*, vol. 14, no. 7, pp. 12-45, 2026.
- [4] S. Sharma, S. Sharma, and A. Athaiya, "Activation Functions in Neural Networks," *International Journal of Engineering Applied Sciences and Technology*, vol. 4, no. 12, pp. 310-316, 2020, doi:10.33564/IJEAST.2020.v04i12.054
- [5] H. Huang, G. Ho, V. Tang, E. Chow, W. Ip and K. Yung, "A lightweight fully convolutional neural network for efficient and accurate tunnel crack detection," *Journal of Advanced Research*, pp. 1-15, 2025.
- [6] A. Oğuz and Ö. F. Ertuğrul, "Chapter 1 - Introduction to deep learning and diagnosis in medicine," in *Diagnostic Biomedical Signal and Image Processing Applications with Deep Learning Methods*, Intelligent Data-Centric Systems, 2023, pp. 1-40.
- [7] A. Dastgir, W. Bin, M. U. Saeed, J. Sheng, L. Site and H. Hassan, "Attention LinkNet-152: a novel encoder-decoder based deep learning network for automated spine segmentation," *Scientific Reports*, pp. 1-13, 2025.
- [8] Y. Chen, X. Zhu, Y. Li, Y. Wei and L. Ye, "Enhanced semantic feature pyramid network for small object detection," *Signal Processing: Image Communication*, vol. 113, pp. 1-10, 2023.
- [9] J. Wan, Z. Qin, X. Cui, F. Yang, M. Yasir, B. Ma and X. Liu, "MBES Seabed Sediment Classification Based on a Decision Fusion Method Using Deep Learning Model," *Remote Sensing*, vol. 14, no. 3708, pp. 1-22, 2022.
- [10] R. V. Poursorkhabi, "Data-driven study of flow & sediment in circular channels using GPR & ANFIS," *Multiscale and Multidisciplinary Modeling, Experiments and Design*, pp. 1-22, 2026.
- [11] Q. Zhu, Y. Zheng, Y. Jiang and J. Yang, "Efficient Multi-Class Semantic Segmentation of High Resolution Aerial Imagery With Dilated Linknet," in *IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)*, 2019.
- [12] M. Wu, Z. Shu, J. Zhang and X. Hu, "HRLINKNET: Linknet With High-Resolution Representation for High-Resolution Satellite Imagery," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021.
- [13] M. P. M. M, V. Mehrotra, S. Aiyar, U. Verma and R. M. Pai, "Automatic Segmentation of River and Land in SAR Images: A Deep Learning Approach," in *2019 IEEE Second International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*, 2019.
- [14] 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," *Applied Sciences*, vol. 11, no. 9691, pp. 1-25, 2021.
- [15] M. N. Mahmud, M. K. Osman, A. P. Ismail, F. Ahmad, K. A. Ahmad and A. Ibrahim, "Road Image Segmentation using Unmanned Aerial Vehicle Images and DeepLab V3+ Semantic Segmentation Model," in *2021 11th IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2021)*, Penang, Malaysia, 2021.
- [16] A. Tsellou, K. Moirgiorgou, G. Plokamakis, G. Livanos, K. Kalaitzakis and M. Zervakis, "Aerial video inspection of Greek power lines structures using machine learning techniques," in *2022 IEEE International Conference on Imaging Systems and Techniques (IST)*, Kaohsiung, Taiwan, 2022.
- [17] N. Rania, H. Douzi, L. Yves and T. Sylvie, "Semantic Segmentation of Diabetic Foot Ulcer Images: Dealing with Small Dataset in DL Approaches," *Image and Signal Processing*, pp. 162-169, 2020.