applied sciences

MDPI

Article

Application of Deep Learning Techniques in Water Level Measurement: Combining Improved SegFormer-UNet Model with Virtual Water Gauge

Zhifeng Xie, Jianhui Jin *, Jianping Wang, Rongxing Zhang and Shenghong Li

Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China; zhifengxie@stu.kust.edu.cn (Z.X.); wjp@kust.edu.cn (J.W.); rongxing4136@foxmail.com (R.Z.); 15987924370@163.com (S.L.)

* Correspondence: jjhkm163@163.com

Abstract: Most computer vision algorithms for water level measurement rely on a physical water gauge in the image, which can pose challenges when the gauge is partially or fully obscured. To overcome this issue, we propose a novel method that combines semantic segmentation with a virtual water gauge. Initially, we compute the perspective transformation matrix between the pixel coordinate system and the virtual water gauge coordinate system based on the projection relationship. We then use an improved SegFormer-UNet segmentation network to accurately segment the water body and background in the image, and determine the water level line based on their boundaries. Finally, we transform the water level line from the pixel coordinate system to the virtual gauge coordinate system using the perspective transformation matrix to obtain the final water level value. Experimental results show that the improved SegFormer-UNet segmentation network achieves an average pixel accuracy of 99.10% and an Intersection Over Union of 98.34%. Field tests confirm that the proposed method can accurately measure the water level with an error of less than 1 cm, meeting the practical application requirements.

Keywords: water level measurement; water level line detection; virtual water gauge; perspective transformation; semantic segmentation

Citation: Xie, Z.; Jin, J.; Wang, J.; Zhang, R.; Li, S. Application of Deep Learning Techniques in Water Level Measurement: Combining Improved SegFormer-UNet Model with Virtual Water Gauge. *Appl. Sci.* **2023**, *13*, 5614. https://doi.org/10.3390/app13095614

Academic Editor: Bing Li

Received: 29 March 2023
Revised: 29 April 2023
Accepted: 30 April 2023
Published: 2 May 2023

1. Introduction

Water level monitoring is a crucial task in hydrological observation, as it is a prerequisite for the effective management of water resources in all forms [1]. Water level changes serve as a valuable indicator of hydrological conditions within a basin, reflecting variations in water storage and flow magnitude. These data are critical for hydrologic forecasting, flood scheduling, and water management as they provide real-time, reliable hydrologic information. For instance, when water levels begin to rise, this may signify an abundance of rainfall or snowmelt in the basin, potentially leading to flooding. Consequently, hydrologists can predict potential flood events based on water level changes and take appropriate measures to minimize the damage caused by floods. Furthermore, water level changes can be used to monitor alterations in water levels within reservoirs, lakes, and rivers, ensuring the effective management of water resources. Thus, the monitoring and analysis of water level changes are of utmost significance to the fields of hydrology and water resources management.

The two primary conventional techniques for determining the water level are manual reading of the water gauge and using a water level meter [2]. However, these methods have limitations, such as limited automation and the difficulty of achieving real-time, high-precision monitoring [3]. Recently, the convergence of network communication technology and computer vision technology has enabled the emergence of computer vision-based water level measurement methods [4]. The proliferation of video monitoring equipment at major hydrographic measurement sites due to advances in network communication

technology provides a strong foundation for this method. The development of computer vision technology further supports the implementation of computer vision-based water level measurement. Compared to traditional water level measurement methods, computer vision-based methods have more potential for development. These methods can utilize existing video monitoring equipment at hydrological stations to obtain images and to calculate water levels using algorithms without the need to purchase additional water level measurement equipment. Moreover, computer vision-based water level measurement methods are low-cost, making them highly desirable for study and application. Water level measurement methods based on computer vision can be categorized into two types, namely those based on digital image processing technology and those based on deep learning technology, depending on the technical approach employed. These two techniques have gained considerable attention by enabling water level measurement with the ability to monitor in real-time and with high precision, which can help overcome the limitations of traditional methods such as the manual reading of water gauges and using water level meters. With advancements in computer vision technology, it is becoming increasingly feasible to implement such techniques in practical applications. The development of more advanced and accurate computer vision-based methods for water level monitoring can have a significant impact on various fields, including hydrology, water resource management, and environmental monitoring.

The current research on computer vision-based water level measurement methods heavily relies on water gauge information, which can be unreliable in practical applications due to corrosion, stains, or obscuration caused by the field environment, or a lack of maintenance. To address this issue, this paper proposes a new water level measurement method that uses an improved SegFormer-UNet model and a virtual water gauge. This method first transforms the water level measurement problem into a water body segmentation problem by introducing a virtual water gauge to calculate the mapping relationship between the virtual water gauge and the actual water gauge in the image. The improved SegFormer-UNet model is then used to segment the water body and obtain the pixel coordinates of the water level line. Finally, the pixel coordinates are converted to coordinates in the virtual water gauge plane to calculate the water level value. The main contributions of this paper are:

(1) The proposal of a new water level measurement method that avoids the flaws of traditional computer vision water level detection methods and achieves ruler-free measurement after calibration when the camera imaging angle is fixed;
(2) The proposal of a water segmentation model based on an improved SegFormer-UNet that achieves better results in the water segmentation task;
(3) The use of the water segmentation model to obtain the pixel coordinates of the water level line directly from the segmentation result, reducing algorithm complexity and enhancing real-time performance.

The remaining sections of this paper are outlined below: Section 2 provides a comprehensive overview of water level detection algorithms, including both traditional methods and image-based techniques. In Section 3, the main methodologies employed in this paper are detailed. Section 4 presents the specific implementation principles of the water level detection algorithms adopted in this paper. Section 5 provides an in-depth analysis of the performance of the proposed water level detection algorithm, supported by field test results. Finally, Section 6 summarizes the key findings and achievements of this paper.

2. Related Works

Water level, which refers to the height of the free horizontal surface of rivers, lakes, and other water bodies relative to a reference plane, has been measured since ancient times. In the past, people used simple wooden poles and ropes to measure water levels, but this method proved imprecise and difficult to apply to large-scale measurements. In the 18th century, European scientists explored the use of barometers for measuring water level, but this approach was limited to static water environments and lost accuracy under fluctuating water conditions. In the early 20th century, hydrology research gained

widespread attention, leading to the invention of new water level measurement techniques. To date, numerous traditional methods for water level measurement have been developed, with many mature techniques available for application. An analysis of past literature reveals that traditional methods for water level measurement at hydrological stations mainly comprise manual observation and water level meter measurement [5]. Manual observation is one of the earliest water level measurement methods, while more advanced water level measurement meters have been developed. Currently, the water level measurement meter utilized at hydrological stations includes a variety of types, such as water pressure [6], radar [7], ultrasonic [8], and laser [9] water level meters.

Although numerous traditional methods for water level detection exist, the trend for modern water level detection technologies is towards digitalization and intelligence. In recent years, automatic water level monitoring systems have gained widespread attention, utilizing artificial intelligence and computer vision technology. High-definition cameras and advanced image processing algorithms allow these systems to automatically identify and analyze water level information, achieving the real-time monitoring and data analysis of water levels. Furthermore, with the development of the Internet and mobile communication technologies, water level measurement applications based on mobile devices and the Internet are becoming increasingly popular [10].

With the development of modern water level detection technology, there is a substantial amount of ongoing research on water level measuring technologies based on computer vision. For example, Liu et al. [11] proposed segmenting and binarizing the water gauge using the water gauge color information, detecting the water level line location using the variance mean threshold approach, and then projecting the water level value using the projection relationship. Chan et al. [12] calculated the correlation coefficient of the same rectangular region in two successive water level pictures to determine the water level line location. Kim et al. [13] initially searched for the region of interest, then used the histogram projection technique to locate the water level line pixel position, and lastly translated the water level line pixel location to the actual water level value using the water scale pixel mapping table. Sun et al. [14] used the water gauge characteristics for edge recognition and keyword localization to compute the water surface height. Several of the above-water-level measurement methods are based on traditional image processing, whereas there are now numerous water level measurement methods based on deep neural networks. For example, Cheng et al. [15] directly used the UNet model for water level line detection, thanks to extensive research on deep learning algorithms. Wang et al. [16] utilized the YOLO (You Only Look Once) v3 network to detect the water gauge, and the ResNet network to conduct scale identification to obtain the water level measurement. Ma et al. [17] proposed that the maximum inter-class variance method combined with morphological processing be used to first extract the rectangle with the smallest side length containing the part of the water gauge, and then calculated the water level value by detecting the area above the water surface in this rectangle area with the YOLOv4 algorithm. Zhang et al. [18] used YOLOv4 to locate the E character of the water gauge and segmented the small area of the water gauge near the water body, then used the DeepLabv3+ algorithm to segment the small area to obtain the water level line and calculate the water level value using linear interpolation.

Despite the variety of methods available for water level detection, each has its own drawbacks. Traditional methods require a significant amount of human and material resources, while modern techniques rely heavily on water gauge features in images. The method presented in this paper aims to address these limitations, offering a solution to improve upon current water level detection approaches.

3. Methodology

3.1. Image Segmentation

Image segmentation, a critical problem in computer vision, is frequently viewed as a subset of the picture classification task. Image segmentation, as opposed to image classification, aims to categorize pixel points inside an image, i.e., segmenting an image

into areas based on the class to which the pixels in the image belong. Traditional image segmentation mainly includes pixel-based thresholding, and region-based, edge-based, and clustering-based methods [19]. These traditional image segmentation methods all have certain limitations in terms of segmentation effectiveness. For example, pixel-based thresholding methods are usually sensitive to lighting changes and noise in the image; region-based segmentation methods are more sensitive to texture information in the image; edge-based methods are sensitive to noise information in the image; and clustering-based methods are sensitive to grey-scale information in the image. In recent years, with the further development of computer vision, image segmentation methods based on deep learning have achieved better results. At present, the mainstream traditional image segmentation methods are usually used in image pre-processing or post-processing work due to the limited use of scenarios. Deep neural networks in the field of image segmentation have achieved many effective results so far. For example, the best-known network model among the early deep neural network-based image segmentation methods is the Fully Convolutional Network (FCN) [20], which segments an image into multiple pixel regions via a fully convolutional network. The FCN had certain shortcomings at the outset of its design, so a series of subsequent improvements emerged, such as the UNet [21], DeepLab series [22–24], etc. The above are all based on the traditional convolutional neural network architecture. With the introduction of the Transformer-based Visual Transformer (ViT) [25] architecture, many other Transformer-based architectures have also emerged in the field of image segmentation in recent years, such as Segmentation Transformer (SETR) [26] and the SegFormer [27] referenced in this paper.

3.2. SegFormer Network Model

The SegFormer network model is a simple and efficient semantic segmentation model recently proposed in the field of image segmentation, using the same encoder–decoder structure as the traditional segmentation model [22]. Unlike traditional convolutional neural network-based segmentation models, the encoder of the SegFormer network model adopts the prevailing Transformer structure. The encoder body consists of four stages, each made of a series of Overlap Patch Embeddings (OPE) layers, Efficient Multihead Self-Attention (EMSA) layers, and Mix Feed Forward (MFFN) layers [27]. The overlap patch embeddings layer is mainly used to reduce the spatial resolution of the front feature map and convert the two-dimensional information in the feature map space into the one-dimensional features required by the Efficient Multihead Self-Attention layer. The latter computes the high-dimensional representation of these features via self-attentiveness, while the final Mix Feed Forward layer is mainly used to enhance the expression of the features. Because the Transformer-based encoder has a larger perceptual field than the traditional convolutional neural network-based encoder, the decoder of the SegFormer network model consists of only a series of Multilayer Perceptron (MLP), and its workflow is divided into four steps. First, the channels of multi-level feature information output from the encoder are adjusted to the same dimensionality via an MLP layer. Then, a linear interpolation algorithm is used to uniformly upsample the spatial dimension of the multi-level feature information to one-quarter of the original image size and stitch it into the channel dimension. In the third step, the stitched features are then fused with information using one more MLP layer. In the final step, the fused features are used for the prediction of the final segmentation mask via another MLP. The structure of the SegFormer network model is shown in Figure 1.

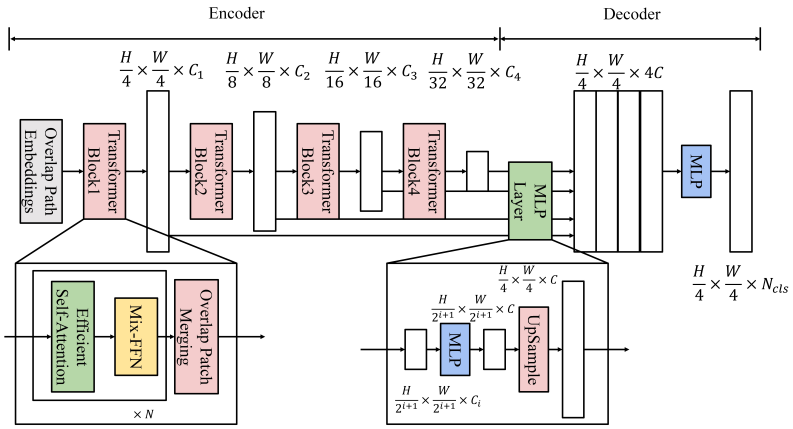

Figure 1. SegFormer network structure.

4. Principle and Implementation

4.1. Water Gauge Mapping Relationship Establishment

Three dimensions of coordinate information are usually required to describe the position of an object in real space, and the image of the water gauge taken by the camera is a projection of the water gauge in the world coordinate system onto the two-dimensional plane of the pixel coordinate system. Considering the installation of the water gauge in the field, it is assumed in the text that the plane of the gauge is approximately perpendicular to the horizontal plane. In a realistic camera set-up scenario, the imaging plane of the camera will usually show a non-parallel relationship with the plane of the water gauge. According to photogrammetry principles, this results in a non-linear projection distortion of the camera image. In order to reduce the errors caused by the projection distortion, this paper uses the perspective transformation [28] to map the water gauge image into a virtual water gauge plane that is parallel to the water gauge plane in the world coordinate system. According to the projection relationship, at this point there is only a proportional relationship between the water gauge in the real water gauge plane and the virtual water gauge plane without any non-linear projection distortion. In order to establish the mapping relationship, three steps are needed and will be presented in the next paragraphs.

Firstly, the matching projection points are selected. The selection of matching projection points is usually arbitrary in the establishment of the perspective relationship, but according to the principles of perspective transformation, at least four pairs of matching projection points must be selected, and the projection points in each coordinate system need to satisfy that three points are not on the same horizontal line [11]. To further ensure the accuracy of the mapping relationship, six pairs of matching projection points are adopted in this paper. The matching projection points are: selected using the rule that they must correspond to the four internal corner points of the E character located at the four corners of the water gauge, as well as the two internal corner points of the E character at any position in the middle of the water gauge, as shown in Figure 2. Let the projection point coordinates in the pixel coordinate system be (x_i, y_i), and the projection point coordinates in the virtual water ruler coordinate system be (u_i, v_i), where $i = 1, 2, 3, 4, 5, 6$.

Next, the perspective matrix is calculated. Let the perspective matrix be $M_{3\times3}$, and according to the perspective projection relationship, we can obtain the coordinates (u_i, v_i) under the plane coordinate system of the virtual water gauge and the coordinates (x_i, y_i) under the pixel coordinate system, as shown in Equation (1) .

$$\begin{cases} u = \frac{m_{11}x+m_{12}y+m_{13}}{m_{31}x+m_{32}y+m_{33}} \\ v = \frac{m_{21}x+m_{22}y+m_{23}}{m_{31}x+m_{32}y+m_{33}} \end{cases} \tag{1}$$

where m_{ij} is the element of the i-th row and j-th column of the perspective matrix. By substituting the six pairs of matching projection points into the above Equation (1), we can

obtain the matrix of equations as in Equation (2). By solving it, we obtain the perspective matrix $M_{3\times3}$.

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ v_1 \\ v_2 \\ v_3 \\ v_4 \\ v_5 \\ v_6 \end{bmatrix} = \begin{bmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -u_1x_1 & -u_1y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -u_2x_2 & -u_2y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -u_3x_3 & -u_3y_3 \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -u_4x_4 & -u_4y_4 \\ x_5 & y_5 & 1 & 0 & 0 & 0 & -u_5x_5 & -u_5y_5 \\ x_6 & y_6 & 1 & 0 & 0 & 0 & -u_6x_6 & -u_6y_6 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -v_1x_1 & -v_1y_1 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -v_2x_2 & -v_2y_2 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -v_3x_3 & -v_3y_3 \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -v_4x_4 & -v_4y_4 \\ 0 & 0 & 0 & x_5 & y_5 & 1 & -v_5x_5 & -v_5y_5 \\ 0 & 0 & 0 & x_6 & y_6 & 1 & -v_6x_6 & -v_6y_6 \end{bmatrix} \cdot \begin{bmatrix} m_{11} \\ m_{12} \\ m_{13} \\ m_{21} \\ m_{22} \\ m_{23} \\ m_{31} \\ m_{32} \end{bmatrix}. \quad (2)$$

Finally, the mapping relationships are calculated. With the perspective matrix $M_{3\times3}$ obtained in the previous step, the coordinates (u_i, v_i) of the water gauge in the virtual gauge coordinate system can be found. Since the real gauge plane scales with the objects in the virtual gauge plane, the coordinates in the virtual gauge coordinate system can be linearly varied to find the water level value.

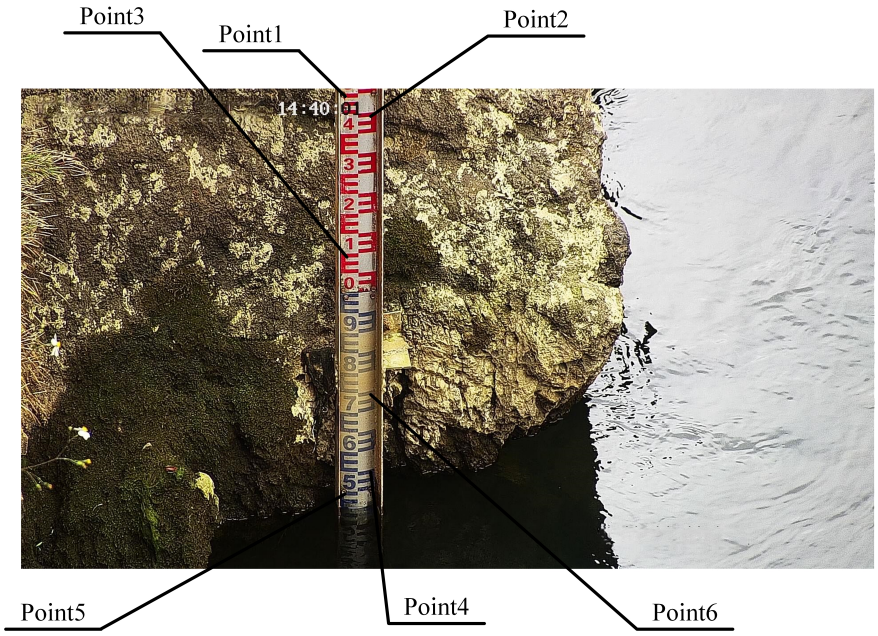

Figure 2. Matching projection point selection.

4.2. Water Segmentation Model

4.2.1. SegFormer-UNet Network Structure

The SegFormer-UNet is a network architecture developed in this paper specifically for segmenting bodies of water in water gauge images. This architecture incorporates the strengths of various semantic segmentation network structures. By summarizing the previous work [20–27], it was discovered that a high-performing semantic segmentation network tends to have the following characteristics: firstly, a strong backbone network as an encoder is a prerequisite. The main reason for the performance improvement of Transformer-based networks over traditional convolutional neural networks is that the Transformer has a stronger encoding capability. Secondly, the network structure needs to have the ability to interact with information at multiple scales. Finally, the network needs to have sufficient spatial perception capability. Based on the above, this paper redesigns

an efficient encoder–decoder architecture for semantic segmentation with reference to the architectures of the SegFormer and UNet networks. The resulting architecture, named SegFormer-UNet, is depicted in Figure 3. The SegFormer-UNet architecture incorporates many features from both SegFormer and UNet, including the attention mechanism from SegFormer. The attention mechanism used in the SegFormer encoder allows the model to focus on key feature areas, improving segmentation accuracy. The skip connections from UNet enable the combination of low-level and high-level feature maps, helping the model to better learn features at multiple scales and improve segmentation robustness. Additionally, the SegFormer-UNet architecture includes Transformer Encoder layers in the encoder, which can help the model capture global contextual information and improve segmentation accuracy. Overall, the advantages of the SegFormer-UNet architecture include focused attention on key feature areas, learning features at multiple scales, and the consideration of global contextual information.

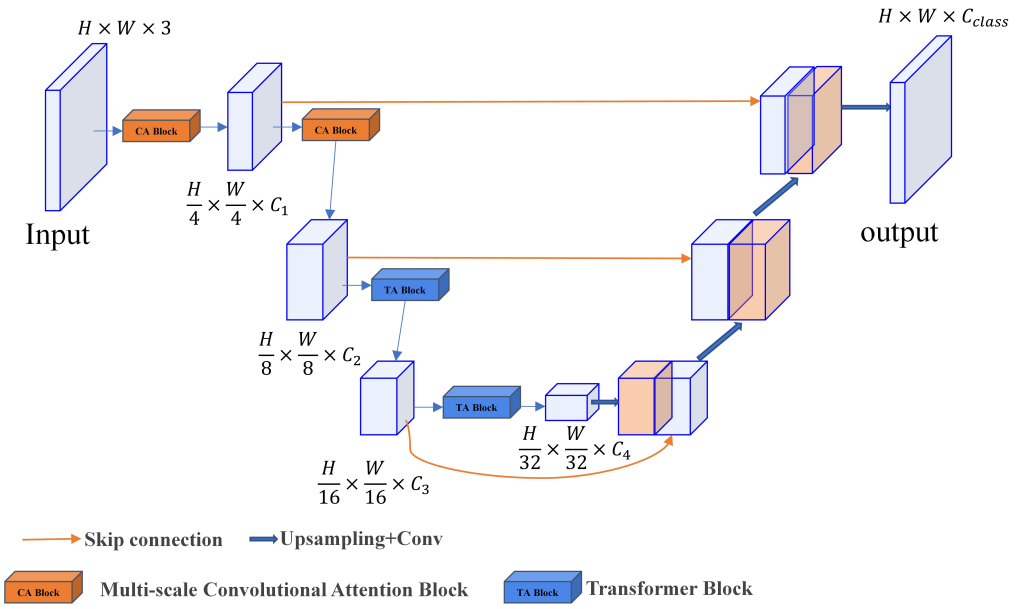

Figure 3. SegFormer_UNet module.

4.2.2. Encoder Design

Transformer models are widely recognized as a highly successful approach in various fields, thanks to their superior performance and robustness [25]. In recent years, a growing number of Visual Transformer (ViT)-based models have emerged in the field of computer vision, indicating a shift in focus from traditional convolutional network architectures [20–24] to Transformer-based model architectures. Although the Transformer architecture is inherently capable of extracting global features using attention mechanisms, it lacks the inductive bias that traditional convolutional networks have. (Inductive bias refers to the inherent assumptions or biases that a machine learning algorithm is built upon. Essentially, it is the prior knowledge that the algorithm uses to make predictions based on new data.) This makes the Transformer-based network model weaker than the traditional convolutional neural network architecture in capturing local relationships, which is detrimental to the detection of edge locations in the water region in this paper. Network models built entirely from Transformers tend to perform well in large-scale datasets, which means that a large amount of data needs to be collected and labeled upfront, while the training of the model is equally costly in terms of computational resources. This is clearly not the best option for this paper, which requires application in a specific production environment. In order to achieve good results on small datasets, the original SegFormer structure is improved by considering a fusion of traditional convolution and self-attentiveness to make full use of the properties of both.

Based on the above considerations, the encoder designed in this paper adopts a hierarchical pyramid structure, with the encoder body consisting of two types of modules: the Multiscale Convolutional Attention Block and the Transformer Block. A series of SegFormer-UNet encoders are designed with reference to the original SegFormer network structure, which are named CTA-B0, CTA-B1, CTA-B2, and CTA-B3 in this paper, respectively. The dimensions of the encoder are designed by referring to the parameters of the encoder as part of the original SegFormer network structure, and customizing and optimizing them for the actual task of this paper. Table 1 shows the composition and specific parameters of each encoder building block. In the table, K_i denotes the patch size of the overlapping patch embedding in Stage i, S_i denotes the stride of the overlapping patch embedding in Stage i, P_i denotes the padding size of the overlapping patch embedding in Stage i, C_i denotes the channel number of the output of Stage i, L_i denotes the number of encoder layers in Stage i, R_i denotes the reduction ratio of the Efficient Self-Attention in Stage i, N_i denotes the head number of the Efficient Self-Attention in Stage i, and E_i denotes the expansion ratio of the feed-forward layer in Stage i. In order to make full use of the detailed features provided by the high-resolution feature map and the advanced semantic features provided by the low-resolution feature map, the encoder in this paper is divided into four stages. At the beginning of each stage, it is necessary to first obtain the features F_i of the previous stage using Overlap Patch Embeddings layers. Specifically, suppose that the number of image features is $H \times W \times 3$, then the number of output features of each stage is $\frac{H}{2^{i+1}} \times \frac{W}{2^{i+1}} \times C_i$, where $i \in \{1, 2, 3, 4\}$. Usually, $C_{i+1} > C_i$.

The patch embedding module in this paper is different from the one commonly used in Transformer, but it is similar to the downsampling process in convolutional neural networks. The specific implementation is to use a stepwise convolution with overlapping regions to perform convolutional operations on the feature map. Except for the first stage where the convolution parameter is set to $K = 7, S = 4, P = 3$, the other three stages are uniformly set to $K = 3, S = 2, P = 1$.

Table 1. Detailed settings of backbone.

Stages	Output Size	Layer Name	CTA-B0	CTA-B1	CTA-B2	CTA-B3
1	$\frac{H}{4} \times \frac{W}{4}$	Overlapping Path Embedding	$K_1 = 7; S_1 = 4; P_1 = 3$			
			$C_1 = 32$		$C_1 = 64$	
		Multiscale Convolutional Attention Encoder	$L_1 = 3$	$L_1 = 2$	$L_1 = 3$	$L_1 = 3$
2	$\frac{H}{8} \times \frac{W}{8}$	Overlapping Path Embedding	$K_2 = 3; S_2 = 2; P_2 = 1$			
			$C_2 = 64$		$C_2 = 128$	
		Multiscale Convolutional Attention Encoder	$L_2 = 3$	$L_2 = 2$	$L_2 = 3$	$L_2 = 5$
3	$\frac{H}{16} \times \frac{W}{16}$	Overlapping Path Embedding	$K_3 = 3; S_3 = 2; P_3 = 1$			
			$C_3 = 160$		$C_3 = 320$	
		Transformer Encoder	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 2$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 2$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 6$	$R_3 = 2$ $N_3 = 5$ $E_3 = 4$ $L_3 = 18$
4	$\frac{H}{32} \times \frac{W}{32}$	Overlapping Path Embedding	$K_4 = 3; S_4 = 2; P_4 = 1$			
			$C_4 = 256$		$C_4 = 512$	
		Transformer Encoder	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 2$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 2$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$	$R_4 = 1$ $N_4 = 8$ $E_4 = 4$ $L_4 = 3$

The multiscale convolutional attention block [29], shown in Figure 4a, is a core module in the encoder of this paper. Its overall structure is similar to ViT, but instead of using the self-attention mechanism, a multiscale convolutional attention block is redesigned using convolution, as shown in Figure 4b. The multiscale convolutional attention block consists of three parts: a deep convolution for aggregating local information, a multibranch convolution for capturing multiscale contexts, and a convolution for mixing different channel relationships. Mathematically, the multiscale convolutional attention module can be written as Equation (3).

$$\begin{aligned} Att &= Conv_{1\times1}\left(\sum_{i=0}^{3} Scale_i(DWConv(F))\right) \\ out &= Att \otimes F \end{aligned} \tag{3}$$

where out denotes the output feature map, Att denotes the convolutional attention weight map, F denotes the input feature map, $\otimes$ denotes the matrix corresponding position element multiplication, and $DWConv$ denotes the depth separable convolution.

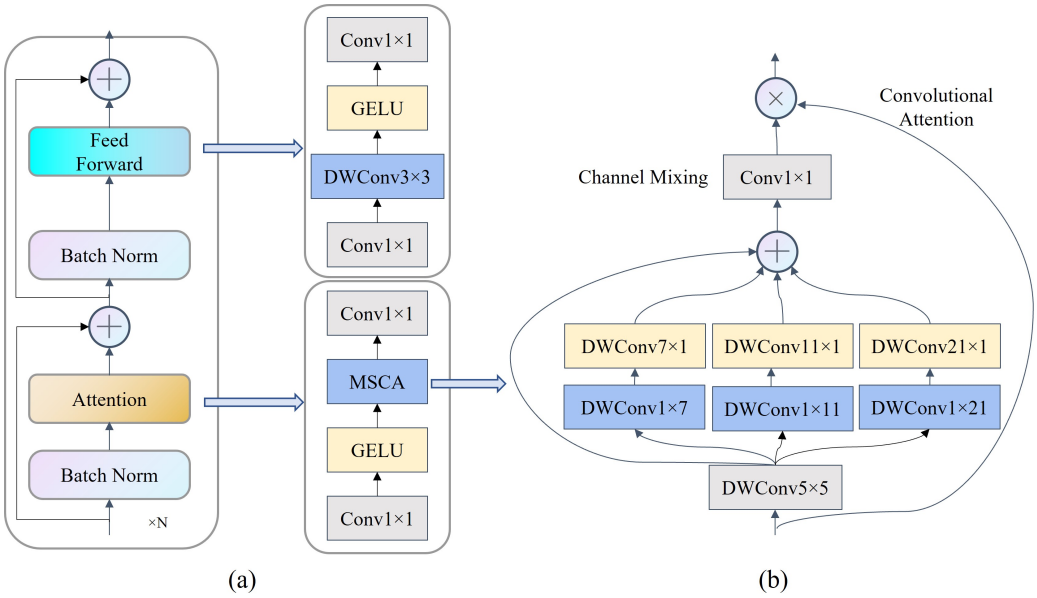

Figure 4. Multi-scale convolutional attention architecture. (**a**) A stage of MSCAN. (**b**) Multi-scale convolutional attention.

Another important module used in this encoder is the existing Transformer block in the original SegFormer, as shown in Figure 5. The core difference between this building block and the traditional Transformer block is the replacement of the multi-headed attention. The efficient self-attentive layer proposed in the original paper is able to reduce the computational complexity, which is implemented by compressing the sequence length using the scaling reduction mentioned in PVT [30], i.e.

$$\begin{aligned} \hat{K} &= \text{Reshape}\left(\frac{N}{R}, C \cdot R\right)(K) \\ K &= \text{Linear}(C \cdot R, C)(\hat{K}) \end{aligned} \tag{4}$$

where K denotes the input sequence, N denotes the sequence length, R denotes the reduction ratio, $\text{Reshape}\left(\frac{N}{R}, C \cdot R\right)(K)$ denotes the deformation of K to $\frac{N}{R} \times (C \cdot R)$, and $\text{Linear}(C \cdot R, C)(\cdot)$ denotes the linear layer with input channel C_{in} and output channel C_{out}.

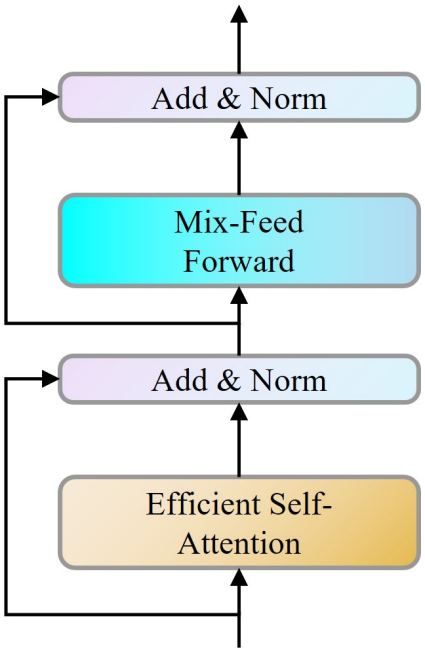

Figure 5. Transformer block structure.

4.2.3. Decoder Design

The decoder structure in the original SegFormer model simply performs upsampling and linear variation on the four different resolutions of the feature maps obtained from the encoder output, and then feeds them into multiple fully connected layers to obtain the final result. Although this lightweight decoder structure avoids excessive computation, the direct and aggressive upsampling of the encoder output may lead to the loss of detailed information, especially in the positioning of the water level line as addressed in this paper. To fully leverage the semantic and spatial information in the feature maps at different stages of the encoder output, this paper proposes a new decoder that references the encoder structure in the UNet network. The structure is illustrated in Figure 6. The decoder employs a skip-connection architecture to make full use of the semantic and spatial information contained in the feature maps at each stage. Meanwhile, the stepwise upsampling technique used in the skip-connection process enables the decoder to capture the relevant features that were lost in the downsampling process of the encoder.

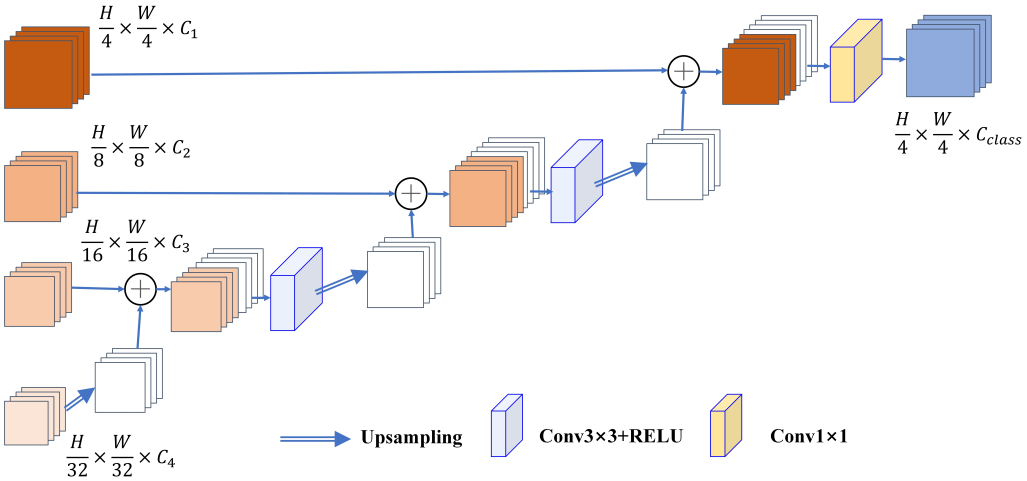

Figure 6. Decoder Module.

4.3. Water Level Line Detection

Once the original water gauge image is fed into the SegFormer-UNet model, the model outputs a mask image containing the segmentation results of the water body and the background. To obtain the position of the water level line in the water gauge image, only the mask image needs to be detected using the Canny edge detection algorithm [31]. The specific detection process is shown in Figure 7.

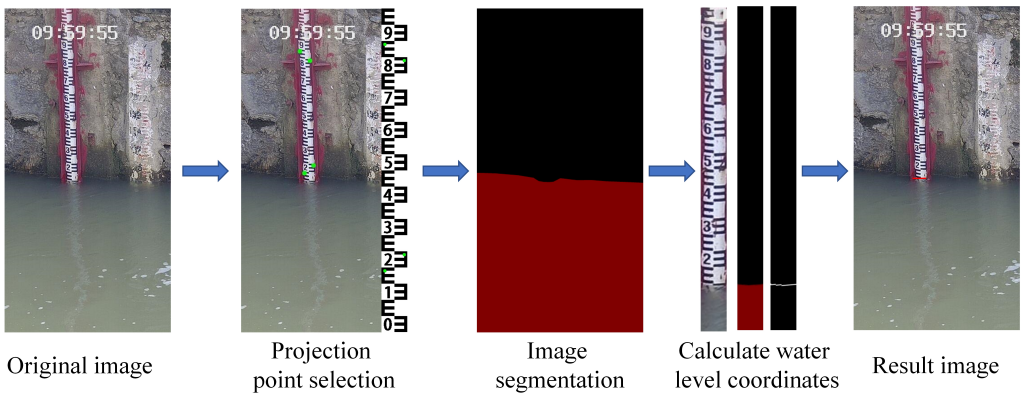

Figure 7. Water level line detection.

4.4. Calculation of Water Level Values

After the location of the water level line is determined, the mapping relationship described in the previous section is utilized to obtain the current water level. Firstly, the perspective transformation matrix $M_{3\times3}$ is employed to convert the water level line from the pixel coordinate system to the virtual gauge coordinate system. The coordinates (u_i, v_i) of the water level line in the virtual gauge coordinate system can then be computed. To obtain a stable water level, the outliers are removed from all the vertical coordinates of the water level in the virtual gauge coordinate system, and the average value is calculated. Finally, the average value is linearly converted to the water level value, according to Equation (5).

$$h = \frac{\sum_{i=0}^{n} v_i}{n} k + b, \tag{5}$$

where h is the water level value, n is the total number of pixel points, v_i is the vertical coordinate value of the i-th point, and k and b are the linear variation scale and offset coefficients, the values of which depend on the precise size of the water gauge in the virtual gauge space chosen. Because the length of a single standard scale in virtual space is 1000 pixels, k and b in this work are -0.1 and 100, respectively. Considering the real-world scenario where two water gauges are joined together vertically, the virtual space water gauge needs to be adjusted accordingly: in this case, as the total length of the virtual space water gauge has been adjusted to 2000 pixels, and the values of k and b in Equation (5) should be set to -0.1 and 200, respectively. After the water level line is converted to the virtual water gauge, assuming its vertical coordinate in the virtual water gauge is v, the water level value can be easily calculated based on v, as shown in Figure 8.

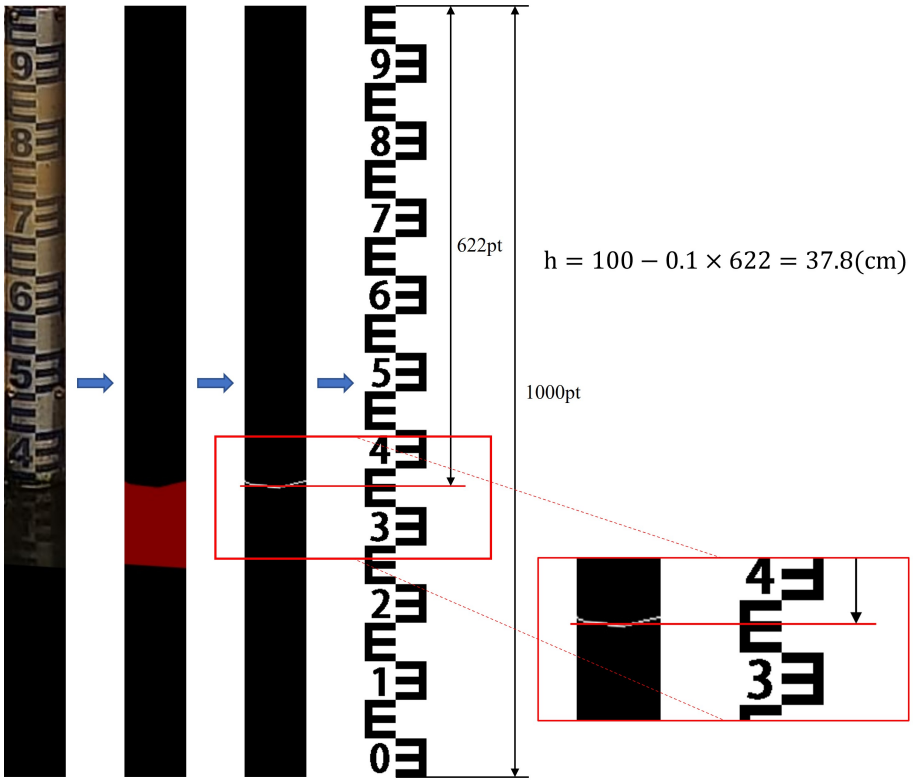

Figure 8. Water level value calculation.

5. Analysis of Results and Field Testing

5.1. Dataset and Experimental Environment

Due to the lack of a publicly available standard water gauge dataset in the field of hydrology, we collected raw data over time from three hydrological stations in the same basin to ensure that the experimental data reflect the real application environment and guarantee the reliability of the model. To ensure the diversity of the data, we randomly extracted 1029 photos containing water gauges from the original data, comprising 833 images in the training set, 93 images in the validation set, and 103 images in the test set. The semantic annotation labels in the dataset were divided into two groups, water body and background, as illustrated in Figure 9.

This paper's experimental platform is Ubuntu 22.04, with the following hardware configuration: Intel(R) Core(TM) i5-12490F CPU and NVIDIA GeForce RTX 3060 12G GPU. The software versions are Python 3.10, Pytorch 1.13, and Cuda 11.7.

To ensure the comparability of the experimental results, uniform hyperparameters are used in the network training process in this paper. The network parameter optimizer used is stochastic gradient descent (SGD) with momentum, where the initial learning rate is set to 1×10^{-2}, the minimum learning rate is 1×10^{-4}, the momentum parameter is set to 0.9, and the weight decay parameter is set to 1×10^{-4}. The learning rate is adjusted using the Cosine Annealing algorithm [32], with the frequency of learning rate adjustment being the number of epochs per training round. For the binary classification task, the Binary Cross Entropy Loss (BCE Loss) combined with the Dice Coefficient Loss is used as the loss function, and its expression is calculated as Equation (6):

$$\begin{aligned} L_{loss} &= L_{BCE} + L_{Dice} \\ L_{BCE} &= -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(\bar{y}_i) + (1 - y_i)\log(1 - \bar{y}_i)] \\ L_{Dice} &= 1 - \frac{2|X \cap Y|}{|X| + |Y|}, \end{aligned} \tag{6}$$

where N denotes the number of all pixel points, y_i denotes the true label value for the i-th point, and $\bar{y}_i$ denotes the predicted label value for the i-th point. X denotes the set of true labels, Y denotes the set of predicted labels, $|X \cap Y|$ is the number of elements of the intersection between labels and predictions, and $|X|$ and $|Y|$ denote the number of elements of labels and predictions, respectively.

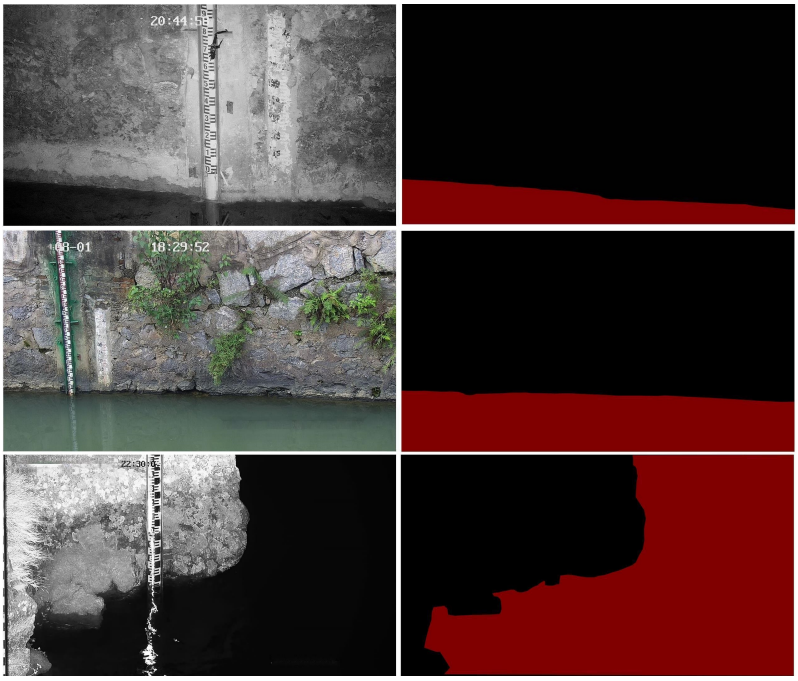

Figure 9. Sample from the dataset.

5.2. Evaluation Indicators

In this paper, we used the mean Intersection over Union (mIoU) and Mean Pixel Accuracy (MPA) as objective metrics to evaluate the performance of the network, which are commonly used in the field of semantic segmentation. Intersection over Union (IoU) represents the ratio of the intersection and union of two sets of pixels of a certain label type to the predicted value, and is a prerequisite for calculating the mean intersection ratio. Its mathematical expression is given by Equation (7):

$$IoU_i = \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{7}$$

The Equation (7) gives the Intersection over Union for each class of labels, and by taking the mean value of each class, $mIoU$ can be obtained; this is calculated as Equation (8). The average pixel accuracy is obtained by averaging the pixel accuracy of each category, the mathematical expression is Equation (9).

$$mIoU = \frac{1}{k+1} \sum_{i=0}^{k} IoU_i \tag{8}$$

$$MPA = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij}}, \tag{9}$$

where IoU_i denotes the intersection ratio of category i, k denotes the total number of categories minus 1, p_{ii} denotes the number of pixel true category i predicted to be category i, p_{ij} denotes the number of pixel true category i predicted to be category j, and p_{ji} denotes the number of pixel true category j predicted to be category i.

In addition to mIoU and mPA, Params and GFLOPs are also commonly used performance metrics for models. Params refers to the number of parameters in the model, while GFLOPs refers to the number of floating-point operations required for inference. These metrics can be used to evaluate model complexity and computational efficiency. When selecting models, it is usually necessary to balance between model accuracy and computational efficiency. Smaller Params and GFLOPs usually mean faster inference speeds and fewer computational resources, but may lead to a decrease in model accuracy. Therefore, Params and GFLOPs should be considered together in the model selection and optimization process to achieve the best performance and efficiency balance.

5.3. Results of The Experiment

5.3.1. Objective Comparison of Segmentation Performance

To test the performance of the SegFormer-UNet segmentation network model proposed in this paper, we compared it with the FCN in [20], the UNet in [21], the DeepLab V3+ in [24], the PSPNet in [33], the HRNet in [34], and the SegFormer in [27]. The evaluation metrics used for comparison are shown in Table 2. It can be observed that the network proposed in this paper achieves an mIoU of 98.34% and an mPA of 99.1%, outperforming the other networks in terms of segmentation performance. These results demonstrate that the SegFormer-UNet segmentation network proposed in this paper is better suited for water body segmentation than other semantic segmentation networks.

Table 2. Comparison results of different networks.

Method	Params (M)	GFLOPs (G)	mIoU (%)	mPA (%)
FCN [20]	32.95	138.86	96.06 (2.28↓)	97.84 (1.26↓)
UNet [21]	24.89	225.84	96.17 (2.17↓)	97.89 (1.21↓)
DeepLab V3+ [24]	54.71	83.42	97.47 (0.87↓)	98.85 (0.25↓)
PSPNet [33]	46.71	59.21	96.66 (1.68↓)	98.17 (0.93↓)
HRNet [34]	65.85	93.83	96.88 (1.46↓)	98.29 (0.81↓)
SegFormer-B5 [27]	84.60	99.75	96.57 (1.77↓)	98.22 (0.88↓)
SegFormer-UNet-B3 (Ours)	46.56	49.43	98.34	99.10

5.3.2. Subjective Comparison of Segmentation Performance

The SegFormer-UNet proposed in this paper demonstrates the best performance in terms of objective evaluation metrics. To establish a perceptual understanding of the segmentation effect of the network model, this paper selects five representative images from the test set for demonstration, as shown in Figure 10. From top to bottom, the images are the original image, the mask image, our model segmentation effect, the FCN segmentation effect, the UNet segmentation effect, the DeepLab V3+ segmentation effect, the PSPNet segmentation effect, the HRNet segmentation effect, and the SegFormer segmentation effect. From the segmentation effect, it can be observed that our model has the best segmentation effect on the edge position.

5.4. Ablation Experiments

5.4.1. Influence of Encoder Size

To verify the rationality of the encoder design in this paper, we conducted tests and analyzed the degree to which model performance is affected by different encoder sizes. The specific test results are shown in Table 3. From the model's performance on the test set, we found that the model's performance improved with increasing encoder size. The largest model of our design, SegFormer-UNet-B3, already outperforms the original SegFormer-B5 model with only half of the parameters and computational effort. This indicates that the encoder design in this paper is reasonable to a great extent.

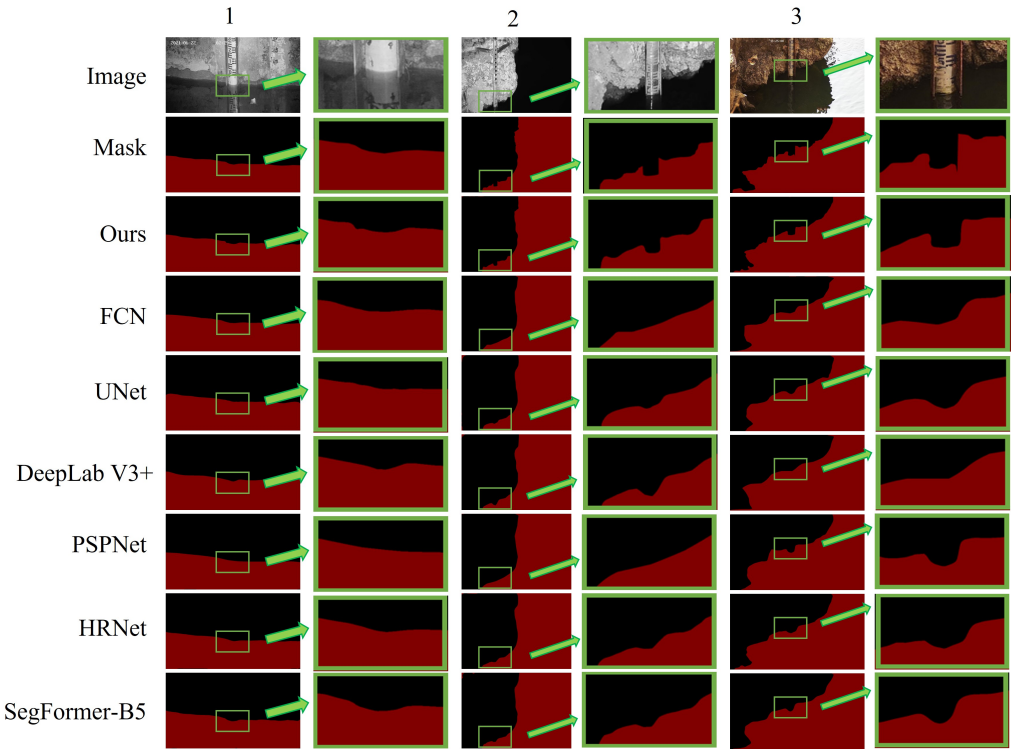

Figure 10. Different network segmentation results.

Table 3. Ablation studies related to Encoder Size.

Method	Backbone	Params (M)	GFLOPs (G)	mIoU (%)	mPA (%)
SegFormer-UNet	B0	5.54	14.49	96.76	98.2
SegFormer	B0	3.72	6.77	93.03	96.04
SegFormer-UNet	B1	15.78	23.50	97.14	98.46
SegFormer	B1	13.28	26.48	95.30	97.55
SegFormer-UNet	B2	26.00	32.04	97.43	98.62
SegFormer	B2	27.35	56.71	96.35	98.12
SegFormer-UNet	B3	46.56	49.43	98.34	99.10
SegFormer	B3	47.22	71.36	96.43	98.10
SegFormer	B4	63.99	85.43	96.30	98.00
SegFormer	B5	84.60	99.76	96.66	98.22

5.4.2. Influence of Encoder Composition Structure

The architecture of the encoder in a neural network model is crucial as it directly impacts the performance of the model. Therefore, designing a robust encoder is necessary to provide better feature information encoding capability to the neural network model. To achieve this, the paper presents a redesigned encoder architecture that integrates convolution and Transformer modules by summarizing previous excellent encoder architectures. This integration enables the neural network model to fully exploit the advantages of both modules. To determine the best organization structure to achieve optimal model performance, we analyzed the impacts of the combined convolution and Transformer module model on model performance magnitude, as shown in Table 4. In the table, CA represents the multi-scale convolutional attention block, and TA represents the Transformer block. From the data in the table, it is evident that the model performance is optimal on the dataset used in this paper when using the CA-CA-TA-TA architecture adopted by the encoder in this paper.

Table 4. Ablation studies related to Encoder Structure.

Method	Architecture				Params (M)	GFLOPs (G)	mIoU (%)	mPA (%)
	Stage 1	Stage 2	Stage 3	Stage 4				
SegFormer-UNet-B0	CA	CA	CA	CA	6.17	15.41	94.26	96.95
	CA	CA	CA	TA	6.24	15.43	96.65	97.56
	CA	CA	TA	TA	5.54	14.49	96.76	98.29
	CA	TA	TA	TA	5.58	14.07	96.69	98.22
SegFormer-UNet-B1	CA	CA	CA	CA	16.76	25.76	95.67	97.73
	CA	CA	CA	TA	17.18	25.87	97.06	98.16
	CA	CA	TA	TA	15.78	23.50	97.14	98.46
	CA	TA	TA	TA	16.30	23.42	97.12	98.42
SegFormer-UNet-B2	CA	CA	CA	CA	29.59	38.97	95.70	97.67
	CA	CA	CA	TA	30.20	39.13	97.16	98.47
	CA	CA	TA	TA	26.00	32.04	97.43	98.62
	CA	TA	TA	TA	26.80	31.90	97.37	98.61
SegFormer-UNet-B3	CA	CA	CA	CA	47.93	59.74	96.21	97.70
	CA	CA	CA	TA	48.55	59.90	97.26	98.57
	CA	CA	TA	TA	46.56	49.43	98.34	99.10
	CA	TA	TA	TA	46.67	46.55	97.27	98.55

5.5. Field Tests

5.5.1. Water Level Line Detection Test and Analysis

This paper utilizes the Average Pixel Absolute Error (APAE) as the evaluation index for water level line detection in order to further validate the effectiveness of the SegFormer-UNet segmentation network. The mathematical expression for APAE is defined as Equation (10).

$$APAE = \frac{1}{n}\sum_{i=0}^{n}|y_i - y_i'|, \tag{10}$$

where n denotes the total number of images tested, y_i denotes the pixel vertical coordinate of the water level line detected by the algorithm for the i-th image, and y_i' denotes the pixel coordinate of the actual water level line in the i-th image calibrated manually.

In this study, four water level detection methods were compared, to quantitatively assess the accuracy of water level line detection, and the results are displayed in Table 5. The comparison of the data in the table shows that this approach has the lowest average absolute pixel error among the five methods, which may avoid the negative impact of water gauge information loss on water level detection. The assessment index demonstrates that SegFormer-UNet can accurately segment the water body and background region in the water gauge images, and that the average absolute pixel error in water level line detection is 1.73 pixels, which is sufficient to meet the criteria for water level line detection accuracy.

Table 5. Average absolute pixel error.

Algorithms	Average Absolute Pixel Error (Pixels)
Wang L [16]	38.32
Lin F [35]	10.47
Liu M [11]	13.06
Zhang R [18]	5.02
Ours	1.73

In Table 5, it is evident that the average absolute pixel error for water level line detection in this paper is significantly lower than those of other water level detection algorithms. This is due to two reasons. Firstly, the water gauge images used in this paper are actual site images, which include a small number of night images and missing water gauge features. Secondly, the proposed algorithm in this paper does not rely on the water gauge itself after establishing the mapping relationship, which substantially eliminates the

negative impact caused by missing water gauge features. In contrast, other water level detection algorithms such as [16] detect the water level line position directly by detecting the lower edge of the water gauge, and [35] uses DeepLabv3+ to directly segment the water gauge, which has a greater accuracy than detecting the lower edge of the water gauge position. The study in [11] uses color information to segment the water level, rendering the algorithm ineffective during the night. The study in [18] can be considered as an improvement on [35], as it crops a small area of the water gauge close to the water surface and then segments the small area to obtain the water level line position. These algorithms rely to some extent on the body characteristics of the water gauge itself, which impact on the accuracy and reliability of the algorithm.

5.5.2. Water Level Measurement Test and Analysis

To quantify the accuracy of the water level measurement algorithm proposed in this paper, we sampled water gauge images from each of the two of the three hydrological stations mentioned earlier over a period of time, took separate water level measurements, and compared them with the results of manual readings. The comparison results are depicted in Figure 11. The error curves for the two stations show that the water level measurement error of the proposed algorithm is within 1 cm, which is sufficiently accurate to meet the requirements for water level measurement.

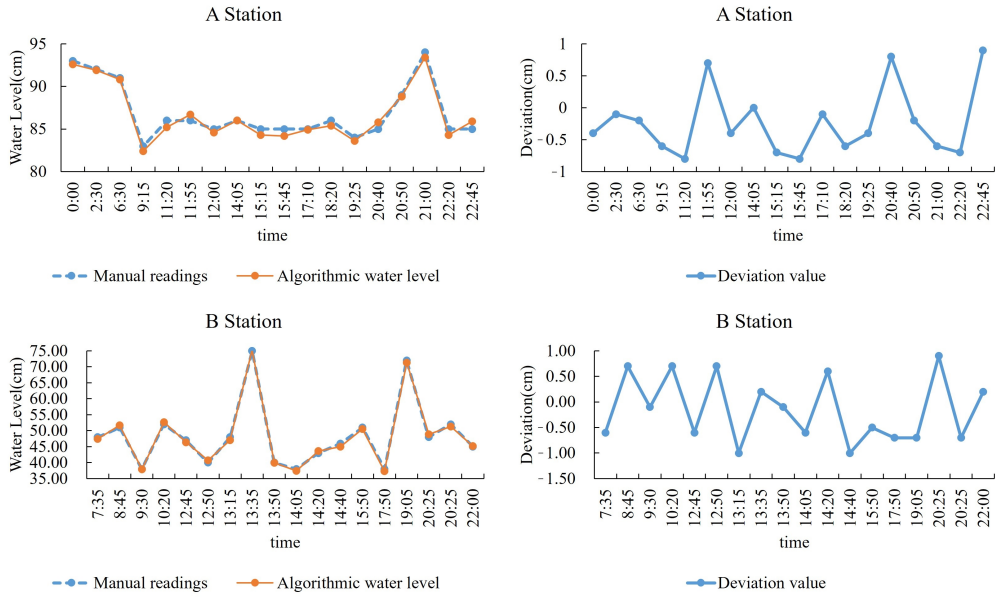

Figure 11. Results of water level measurement.

5.5.3. Measurement Test without a Water Gauge

The primary objective of the algorithm proposed in this paper is to address the limitations of current mainstream water level detection algorithms in field environments where water gauge information may be missing due to uncontrollable conditions such as corrosion, stains, strong reflection, and low brightness, leading to a loss of detection accuracy or complete failure. By establishing a mapping relationship, the algorithm is capable of water gauge-free measurement using a fixed-view camera after the initial calibration. To evaluate the effectiveness of the proposed algorithm in achieving water gauge-free measurement, we conducted simulation experiments as follows: first, we selected a test site with a constant water level; then, we captured images of the water gauge after setting it up and after removing it using a fixed-point camera; finally, we tested the captured images using the proposed algorithm, and compared the water level detection results obtained with and without the water gauge. The test results shown in Figure 12

demonstrate that the proposed algorithm can achieve water gauge-free measurements while maintaining a constant camera shooting angle.

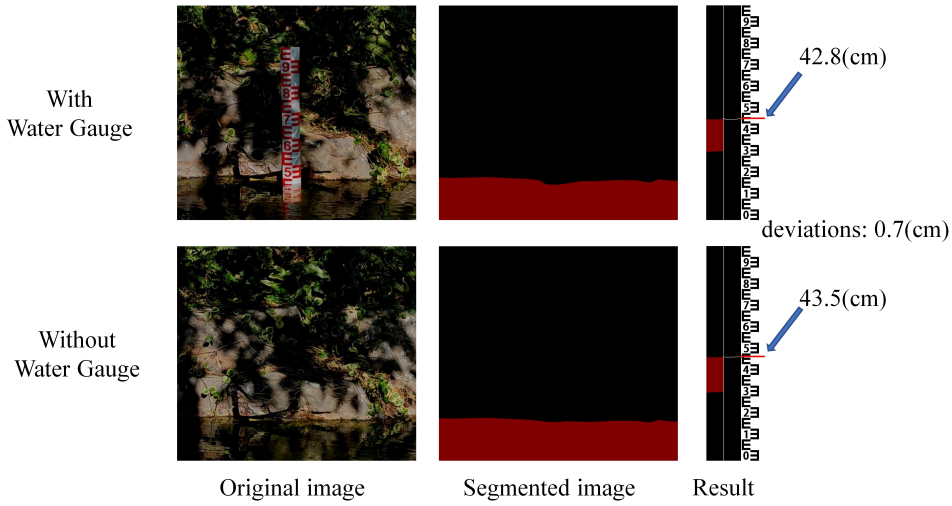

Figure 12. Measurement test without water gauge.

6. Conclusions

This paper presents a novel approach to water level measurement using a combination of improved SegFormer-UNet and a virtual water gauge, which addresses the limitations of previous computer vision-based methods that rely heavily on gauge information. Specifically, we transform the water level measurement problem into a water body segmentation task by establishing a mapping relationship between the water gauge and the image. Our proposed SegFormer-UNet model achieves an impressive average pixel accuracy of 99.10% and an Intersection Over Union of 98.34% on the segmentation index, outperforming the original SegFormer model and other mainstream segmentation networks. Additionally, we demonstrate that our method accurately detects the water level line with an average absolute pixel error of 1.73 pixels compared to the actual water level line. In field tests, our method achieves water level measurements with an average absolute deviation of less than 1 cm, meeting the requirements for practical applications. Notably, we show that our method can measure water levels without relying on a physical gauge while maintaining the same camera imaging angle. To enhance the generalizability of the algorithm, future work will focus on improving the model's generalization capability for different water surface environments, streamlining the model structure for faster inference, and increasing the diversity of the dataset for a thorough validation of the algorithm's generalizability.

Author Contributions: Conceptualization, Z.X., J.J. and J.W.; methodology, J.J.; software, Z.X.; validation, Z.X., J.J., R.Z. and S.L.; formal analysis, J.J.; investigation, J.J.; resources, J.J. and J.W.; writing—original draft preparation, Z.X.; writing—review and editing, J.J.; funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by "Yunnan Xingdian Talents Support Plan" project of Yunnan and Key Projects of Yunnan Basic Research Plan (202101AS070016).

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Muste, M.; Ho, H.C.; Kim, D. Considerations on direct stream flow measurements using video imagery:Outlook and research needs. *J. Hydro Environ. Res.* **2011**, *5*, 289–300. [CrossRef]
2. Zhang, K.; Wang, J.; Zhang, G.; Liu, M. Review of image water level detection. *Electron. Meas. Technol.* **2021**, *44*, 104–113.
3. Zhang, Y. A Brief Discussion on Model Selection of Water Level Gauge for Mountain River. *Water Resour. Informatiz.* **2008**, *4*, 45–46.

4. Cheng, G. *Research on Water Level Scale Recognition Based on Digital Image Processing*; South China University of Technology: Guangzhou, China, 2017.
5. Nie, H.; Liu, K.; Ou, Z. A review of water level measurement methods and equipment. *South. Agric. Mach.* **2020**, *51*, 48–49.
6. Zhang, Y.; Zong, J.; Jiang, D.; Li, S. Design and implementation of bubble pressure type water level meter field detection device. *Hydrology* **2021**, *41*, 60–65.
7. Ma, Y. Analysis of radar water level meter and artificial observation water level ratio at Wushenggong hydrological station. *Groundwater* **2022**, *44*, 205–206.
8. Feng, X. Analysis of ultrasonic water level meter in Shule River irrigation area bucket mouth measurement test application. *Agric. Technol. Inf.* **2020**, *7*, 123–125. 128.
9. Chen, X. Application of laser water level meter in Yantan hydropower station. *Hongshui River* **2015**, *34*, 101–104.
10. Seibert, J.; Strobl, B.; Etter, S.; Hummer, P.; van Meerveld, H.J. Virtual Staff Gauges for Crowd Based Stream Level Observations. *Front. Earth Sci.* **2019**, *7*, 70. [CrossRef]
11. Liu, M.; Che, G.; Zhang, K.; Wang, J.; Ouyang, X. A water level measurement method for indefinite water gauge image. *Chin. J. Sci. Instrum.* **2021**, *42*, 250–258.
12. Lee, C.J.; Seo, M.B.; Kim, D.G.; Kwon, S.I. A novel water surface detection method based on correlation analysis for rectangular control area. *J. Korea Water Resour. Assoc.* **2012**, *45*, 1227–1241. [CrossRef]
13. Kim, J.; Han, Y.; Hahn, H. Image-based water level measurement method under stained ruler. *J. Meas. Sci. Instrum.* **2010**, *1*, 28–31.
14. Sun, W.; Wang, D.; Xu, S.; Wang, J.; Ma, Z. Water Level Detection Algorithm Based on Computer Vision. *J. Appl. Sci.* **2022**, *40*, 434–447.
15. Cheng, S.; Zhao, K.; Zhang, S.; Zhang, D. Water Level Detection Based on U-net. *Acta Metrol. Sin.* **2019**, *40*, 361–366.
16. Wang, L.; Chen, M.; Meng, K. Research on water level recognition method based on deep learning algorithms. *Water Resour. Informatiz.* **2020**, *3*, 39–43.
17. Ma, R.; Zhou, W.; Zou, Y. Water Level Recognition Method Based on Traditional Image Processing Algorithm and YOLOv4. *Comput. Meas. Control* **2022**, *30*, 219–225.
18. Zhang, R.; Zhang, G.; Xie, Z.; Liu, M. Research on water gauge water level detection method under small area guidance. *J. Yunnan Univ.* **2023**, *3*, 1–13.
19. Deeparani, K.; Sudhakar, P. Efficient image segmentation and implementation of K-means clustering. *Mater. Today Proc.* **2021**, *45*, 8076–8079. [CrossRef]
20. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
21. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention, Munich, Germany, 5–9 October 2015; pp. 234–241.
22. Chen, L.C.; Pap, reou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs. *arXiv* **2014**, arXiv:1412.7062.
23. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
24. Chen, L.C.; Zhu, Y.; Pap, reou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Proceedings of the European conference on computer vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
25. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
26. Zheng, S.; Lu, J.; Zhao, H.; Zhu, X.; Luo, Z.; Wang, Y.; Fu, Y.; Feng, J.; Xiang, T.; Torr, P.H.; et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 14–19 June 2021; pp. 6881–6890.
27. Xie, E.; Wang, W.; Yu, Z.; An kumar, A.; Alvarez, J.M.; Luo, P. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 12077–12090.
28. Chen, Z.; Tang, X.; Lin, Z. Research and implementation of adaptive correction and quality enhancement algorithm for distorted image. *J. Comput. Appl.* **2020**, *40*, 180–184.
29. Guo, M.H.; Lu, C.Z.; Hou, Q.; Liu, Z.; Cheng, M.M.; Hu, S.M. Segnext: Rethinking convolutional attention design for semantic segmentation. *arXiv* **2022**, arXiv:2209.08575.
30. Wang, W.; Xie, E.; Li, X.; Fan, D.P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 568–578.
31. Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *6*, 679–698. [CrossRef]
32. Loshchilov, I.; Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv* **2016**, arXiv:1608.03983.
33. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.

34. Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. Deep high-resolution representation learning for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *43*, 3349–3364. [CrossRef]
35. Lin, F.; Yu, Z.; Jin, Q.; You, A. Semantic segmentation and scale recognition based water-level monitoring algorithm. *J. Coast. Res.* **2020**, *105*, 185–189. [CrossRef]