water

MDPI

Article

Application of Closed-Circuit Television Image Segmentation for Irrigation Channel Water Level Measurement

Kwihoon Kim and Jin-Yong Choi *

Department of Rural Systems Engineering, College of Agriculture and Life Sciences, Seoul National University, Seoul 08826, Republic of Korea; kgh0330@snu.ac.kr
* Correspondence: iamchoi@snu.ac.kr

Abstract: Measuring water levels in an irrigation channel is an important task in irrigation system decision making and estimating the quantity of irrigation water supplies. This study aimed to measure water levels with image information from an irrigation channel. Images were obtained from a CCTV (closed-circuit television) camera and manually annotated to create ground-truth mask images. A comparative analysis was performed using four backbone models (ResNet-18, ResNet-50, VGGNet-16, and VGGNet-19) and two segmentation models (U-Net and Link-Net). ROIs (Regions of Interest), mostly related to the water levels, were selected for converting water pixels to water levels. The U-Net with ResNet-50 backbone model outperformed other combinations in terms of the F1 score and robustness, and selecting an ROI and using a quadratic line between water pixels and water levels showed an R^2 of 0.99, MAE (Mean Absolute Error) of 0.01 m, and ME (Maximum Error) of 0.05 m. The F1 score of 313 test datasets was 0.99, indicating that the water surface was sufficiently segmented and the water level measurement errors were within the irrigation system's acceptable range. Although this methodology requires initial work to build the datasets and the model, it enables an accurate and low-cost water level measurement.

Keywords: irrigation channel; semantic segmentation; CCTV image; water level measurement; U-Net; Link-Net

check for updates

Citation: Kim, K.; Choi, J.-Y. Application of Closed-Circuit Television Image Segmentation for Irrigation Channel Water Level Measurement. *Water* **2023**, *15*, 3308. https://doi.org/10.3390/w15183308

Academic Editor: Songhao Shang

Received: 26 August 2023
Revised: 13 September 2023
Accepted: 17 September 2023
Published: 19 September 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Irrigation is an essential agricultural practice, and efficient water use and management are today's major concerns [1]. Of the irrigation practices, surface irrigation is the predominant irrigation practice worldwide owing to its low operation and maintenance costs [2,3]. Nearly 70% of the water consumed in the world is used for irrigation, and most of the water is transported through open channels [4]. The surface irrigation system conveys water through small channels with a gentle slope towards the downstream end [5]. In the basin irrigation system, one of the surface irrigation systems traditionally used to grow rice, earthen banks are usually constructed around the basin, leaving a notch for the water inlet, and small channels are linked to each inlet [5]. Because of this, water losses in irrigation channels are large, but they can be substantially reduced by improving the control systems with appropriate information infrastructure [6].

A single irrigation system generally covers large areas, which often leads to conflicting interests among farmers. It also takes a long time for water to reach or stop at the end, which leads to water use inefficiencies depending on the timing of the water supply decision. The decisions are usually made empirically by experienced farmers or related WUAs (Water User Associations). However, the aging and declining number of farmers are raising concerns that traditional experience-based irrigation is not sustainable [7]. In addition, the development of ICT (Information and Communications Technology) can also fulfill the need to measure the water supply quantity in irrigation channels [8]. These are driving the need for more objective and data-based decisions for irrigation operation and management.

Most previous research has measured water levels by applying ultrasonic sensors in irrigation channels or paddy fields. Lozano [9] measured the water level of the irrigation canal for the application of an automatic control in the main canal, and Masseroni [10] constructed an automatic system to control the gate for maintaining the setup water levels inside the paddy field. Hamdi [11] measured the water level of the irrigation channel to apply IoT (Internet of Things) technology to the water irrigation system. Water levels measured in the channel were also applied to estimate the water supply from an agricultural reservoir [12]. To assess whether the quantity of irrigation water supplied from the agricultural reservoir was adequate compared to the demand, the amount was calculated using measured water levels and a locally derived rating curve [13,14]. Likewise, measured water levels have been used for irrigation system decision making and estimating the quantity of agricultural water supplies. Nevertheless, their use was limited to large-scale main channel measurements as ultrasonic sensors still have disadvantages, such as relatively high costs in installation and maintenance [15–17].

In contrast, CCTV cameras are relatively cost-effective and use visual information [15–17]. Irrigation channel management using CCTV is intuitive, and it is possible to respond immediately when a problem occurs. In addition, image processing techniques have developed rapidly along with the advancement of computing technology and are being actively applied in fields such as remote sensing, autonomous driving, and image diagnosis. Among the image models, semantic segmentation models have been commonly used in water-related research. The performance of segmentation models has improved significantly with the advent of FCN (Fully Convolutional Network) series models, and, for this reason, FCN series models transfer the pretrained classification model as a backbone, and comparison studies have been conducted widely [18,19]. The FCN series models include U-Net, Link-Net, Seg-Net, and DeepLabv3+, and there have been many studies comparing their performances [20–25]. Of the FCN models, U-Net series models have shown slightly better performances, but research on comparing U-Net with Link-Net was not common. Also, it was hard to review research on FCN series segmentation results coupled with backbone models.

There has been research into applying image-processing techniques to rivers and water channels. Hies [26] attempted to detect water levels by applying the edge detection technique to CCTV images of open channels. Lin [27] detected the water levels in a channel using a surveillance camera and a level gauge with edge detection and photogrammetric principles. Both studies applied the edge detection technique to detect water levels in the water channels. However, the line-based method is sensitive to image noises, which means that even small changes in the background are likely to generate frequent or large errors. There have been studies using deep-learning-based models to segment water surfaces. The performance was relatively low in research using urban flood images due to the complexity of various objects [28,29], while research using simple river images showed better performance [30,31]. In water segmentation, simpler images showed better performances, and some studies have tried to use a reference point related to the water level [21,32]. Bai [33] segmented the water surface and calculated water levels with the area of a remaining staff gauge. Kim [34] tried to estimate water levels by correlating the number of water pixels in a column or row with the water levels. Likewise, a wide range of high-quality data can be obtained from irrigation channels by applying deep-learning-based semantic segmentation techniques that perform well on CCTV images.

The objective of this study is to measure water levels in an agricultural channel using CCTV image semantic segmentation. For this purpose, this study compared the segmentation performances of different combinations and developed a method to convert the generated mask images into water levels. U-Net and Link-Net were used as segmentation models, and ResNet-18, ResNet-50, VGGNet-16, and VGGNet-19 were used as backbone models. From the model of the best-performing combination, ROI (Region of Interest) images were applied to compare the estimated water levels with full-resolution images, and different pixel-level conversion equations were used to convert water levels.

2. Materials and Methods

In this study, ultrasonic water levels, measured simultaneously as the image data, were used for the datasets of the models, and Figure 1 shows the study flow. Semantic segmentation requires an annotated mask image for each image, and this study manually annotated mask images using the Apeer program. The performance of segmentation was evaluated with an F1 score. Two models, U-Net and Link-Net, were used to compare the performance of the segmentation model, and four models (ResNet and VGGNet series) were used to compare the performance of backbone models. Then, the configuration that showed the best performance was used for the water segmentation process.

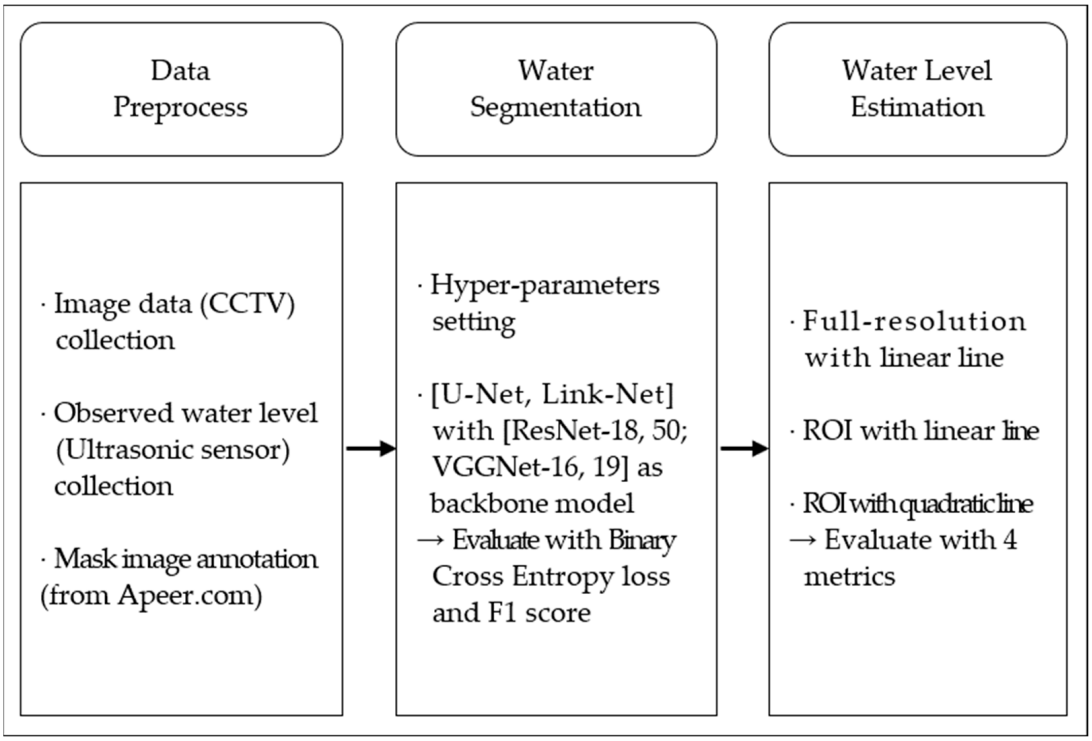


Figure 1. Flow chart of this study (data preprocess, water segmentation, and water level estimation).

The water levels were converted using the pixels of generated mask images and the prederived pixel-to-level equations. In this step, this study developed methods to convert the number of water pixels into the water levels. Firstly, water levels were estimated using the generated mask image's full resolutions (1280 × 720 × 3) and a linear line. Secondly, the ROI was selected to replace the full-resolution images, and a linear line was applied to convert water pixels into water levels. Finally, a quadratic line was derived using the number of pixels in the ROI image, and the estimated water levels were compared with previous results using several metrics.

2.1. Dataset

This study selected an irrigation channel with a CCTV camera and an ultrasonic sensor installed. Images were acquired from the stationary CCTV camera observing an irrigation channel from the same angle. Images were taken every 10 min between 2 June and 6 July 2021, and water levels observed simultaneously from the ultrasonic sensor were matched to each corresponding image. A total of 1564 images were used, with water levels ranging from 0.63 to 1.10 m. As the semantic segmentation model requires a ground-truth mask image for each image, this study utilized the Apeer program. Figure 2 shows example raw images and the generated mask images used in this study. The raw images were captured from the video file and saved in the PNG format, and the generated mask images were saved in the TIFF format. The raw images had a size of 1280 × 720 × 3 pixels, and the

generated mask images had a size of 1280 × 720 × 1. In this study, an ROI was selected for application, and the datasets were reconstructed by cropping 256 × 256 from the full-resolution raw and mask images. The full-resolution image indicates the unprocessed image taken by the camera, whereas the ROI image indicates a specific cropped part of the unprocessed image. Table 1 shows the dataset specifications used in this study.

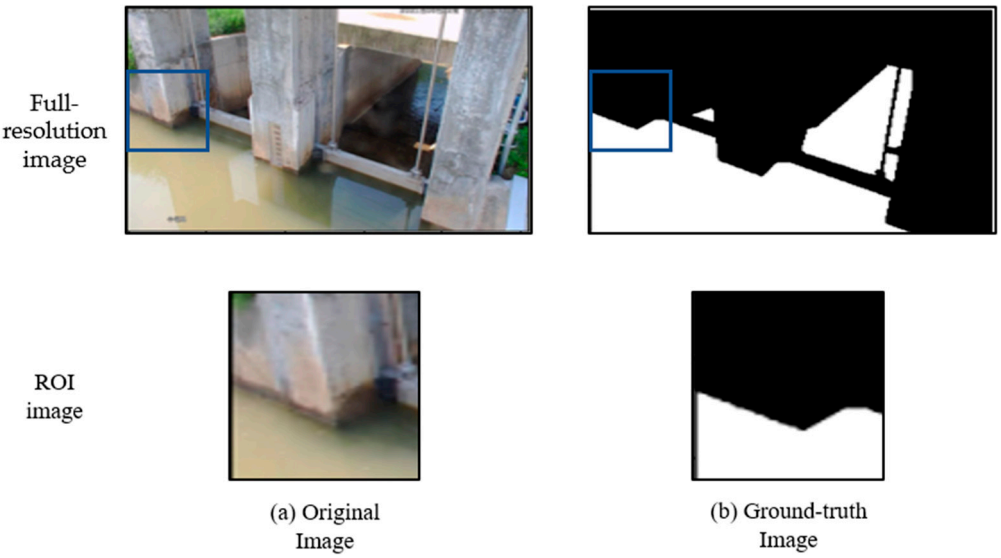


Figure 2. Example images of this study: (**a**) original images; (**b**) associated manually annotated ground-truth images for full resolution and ROI.

Table 1. Dataset specifications used in this study.

Image Type	File Format	Resolution		Number of Datasets			Water Level Range (m)
		Full-Resolution	ROI	Train	Validation	Test	
Raw image	PNG	1280 × 720 × 3	256 × 256 × 3	1125	126	313	0.63~1.10
Mask image	TIFF	1280 × 720 × 1	256 × 256 × 1				

The water levels were measured every 0.01 m from the ultrasonic sensor, with 33 water level values in 1564 images. A total of 33 ground-truth mask images were annotated and allocated to each image having the same water level value. Therefore, a total of 1564 datasets were constructed in this study, of which 70% were used for training, 10% for validation, and 20% for testing. This study used the Train_test_split module from the Scikit-learn package in Python to split training and validating datasets. When splitting the datasets, datasets for the training, validation, and testing were divided randomly.

2.2. Hardware and Software

The platform for constructing the model was a Windows-based computer (64-bit Windows 10 released in 2019) with an Intel(R) Xeon(R) Silver 4214R CPU with 2.40 GHz, 32 GB of RAM, and an NVIDIA Quadro RTX 5000 graphics card with a 513 MHz clock speed. For the programming language, Python 3.6.13 was used, including various libraries. The 2.4.1 version of the Tensorflow-GPU was installed using 'pip install' to avoid compatibility errors. The Nvidia Cuda compiler driver version was 451.48, with the 11.0 toolkit version.

2.3. Segmentation Model Construction (U-Net and Link-Net)

U-Net is a fully convolutional network architecture initially designed to perform binary segmentation of biomedical images [22]. It consists of a contracting path and a

symmetric expanding path. In the contraction path, the spatial information is reduced, while feature information is increased. The expanding path combines the feature and spatial information through a sequence of up-convolutions and concatenations with high-resolution features from the contracting path [35]. The Link-Net architecture is also based on the encoder–decoder structure of the fully convolutional series, and, thereby, it also has a contracting and expanding path, which is characteristic of FCN series models. Link-Net is an efficient semantic segmentation neural network that takes advantage of skip connections and residual blocks. Link-Net initially used ResNet-18 as its encoder, which is a relatively light but outperforming network [20,36]. The Link-Net and U-Net used in this study share the same encoder structure and differ only in their decoder structure. Instead of concatenating layers from the encoder with the decoder as in U-Net, Link-Net adds layers from the encoder to the decoder. For this reason, Link-Net has a smaller number of parameters compared to U-Net, which leads to a relatively shorter computation time.

This study conducted a comparative analysis using U-Net and Link-Net models based on the segmentation performance and the computation time. The hyper-parameters were applied equally, and the encoder weights were pretrained using ImageNet data. The U-Net and Link-Net models were constructed using the open-source library named Segmentation_models, which is mainly based on the Keras library. Comparative experiments were carried out using ResNet-18, ResNet-50, VGGNet-16, and VGGNet-19 as backbone models. After experiments with identically allocated datasets, the best-performing configuration was selected to be applied in the water level estimation step.

The architectures of each combination are shown in Table 2. In the encoding path, each combination used the same structure of each backbone model. The convolution blocks at each stage were stacked by repeating the conv, batch norm, ReLu activation, and zero padding layers as in the block architecture section of Table 2. In the decoding path, each combination was configured with the same structure in U-Net regardless of the backbone model type. Up-sampling and Concatenation were stacked once, and then the 3×3 convolution blocks were stacked twice. In the case of Link-Net, a 1×1 sized convolution block was stacked, followed by up-sampling, and then consecutive 3×3 and 1×1 convolution blocks were stacked with an Add layer. In the decoding path of Link-Net, the number of feature maps differed depending on the backbone model type.

After configuration, each segmentation model was constructed using training datasets, and the validation datasets helped notice whether overfitting occurred. Several hyper-parameters needed to be fixed in the segmentation models, such as the number of epochs, optimizer, batch size, and loss function type. This study set hyper-parameters with a trial-and-error method referencing other previous research. After trials, the configurations yielding the most stable results used an Adam optimizer, batch size of 8, and binary cross-entropy loss function, as shown in Table 3. In order to choose the appropriate epoch for the models, this study referenced the binary cross-entropy loss of the training and validation process. The number of epochs was applied differently depending on the validation losses of each combination. In the testing process, segmented mask images from the model were acquired and compared with the ground-truth mask images.

2.4. Water Level Estimation

The next step was to estimate the water levels from the segmented mask images. This study estimated the water levels using conversion equations and the number of water pixels using train datasets. Two types of regression lines were applied to the water level estimation step, and the results were evaluated with the observed water levels of test datasets. Firstly, linear lines were applied to full-resolution and ROI datasets. The lines were derived using the least square method from the train datasets. The formula for the full-resolution datasets was [$y = x \times 9 \times 10^{-6} - 1.782$], and the formula for the ROI datasets was [$y = x \times 3 \times 10^{-5} + 0.197$]. Secondly, a quadratic line was applied to the ROI datasets. The quadratic line was derived considering the relationship between pixels and levels in train datasets, and the formula for the ROI datasets was [$y = x^2 \times 2 \times 10^{-9} - x \times 3 \times 10^{-5} + 0.785$].

Table 2. Model architecture for each segmentation and backbone model.

Encoder						
Stage	**ResNet-18**	**ResNet-50**	**VGGNet-16**	**VGGNet-19**	**Block Architecture**	
					U-Net	**Link-Net**
Conv1	$[7 \times 7,\ 64] \times 1$	$[7 \times 7,\ 64] \times 1$	$[3 \times 3,\ 64] \times 2$	$[3 \times 3,\ 64] \times 2$	Convolution ↓ BatchNorm ↓ ReLuActivation ↓ Zero Padding	
Conv2	$\begin{bmatrix} 3 \times 3,\ 64 \\ 3 \times 3,\ 64 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1,\ 64 \\ 3 \times 3,\ 64 \\ 1 \times 1,\ 256 \end{bmatrix} \times 3$	$[3 \times 3,\ 128] \times 2$	$[3 \times 3,\ 128] \times 2$		
Conv3	$\begin{bmatrix} 3 \times 3,\ 128 \\ 3 \times 3,\ 128 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1,\ 128 \\ 3 \times 3,\ 128 \\ 1 \times 1,\ 512 \end{bmatrix} \times 4$	$[3 \times 3,\ 256] \times 3$	$[3 \times 3,\ 256] \times 4$		
Conv4	$\begin{bmatrix} 3 \times 3,\ 256 \\ 3 \times 3,\ 256 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1,\ 256 \\ 3 \times 3,\ 256 \\ 1 \times 1,\ 1024 \end{bmatrix} \times 6$	$[3 \times 3,\ 512] \times 3$	$[3 \times 3,\ 512] \times 4$		
Conv5	$\begin{bmatrix} 3 \times 3,\ 512 \\ 3 \times 3,\ 512 \end{bmatrix} \times 2$	$\begin{bmatrix} 1 \times 1,\ 512 \\ 3 \times 3,\ 512 \\ 1 \times 1,\ 2,048 \end{bmatrix} \times 3$	$[3 \times 3,\ 512] \times 3$	$[3 \times 3,\ 512] \times 4$		
Decoder						
Stage	**U-Net**	**Link-Net**			**Block Architecture**	
		ResNet-18	**ResNet-50**	**VGGNet-16 and VGGNet-19**	**U-Net**	**Link-Net**
Conv1	$[3 \times 3,\ 256] \times 2$	$[1 \times 1,\ 128] \times 1$ $[3 \times 3,\ 128] \times 1$ $[1 \times 1,\ 256] \times 1$	$[1 \times 1,\ 512] \times 1$ $[3 \times 3,\ 512] \times 1$ $[1 \times 1,\ 1024] \times 1$	$[1 \times 1,\ 128] \times 1$ $[3 \times 3,\ 128] \times 1$ $[1 \times 1,\ 512] \times 1$	Up-sampling ↓ Concatenation ↓ $\begin{bmatrix} Convolution \\ BatchNorm \\ ReLuActivation \end{bmatrix} \times 2$	$\begin{bmatrix} Convolution \\ BatchNorm \\ ReLuActivation \end{bmatrix} \times 1$ ↓ Up-sampling ↓ $\begin{bmatrix} Convolution \\ BatchNorm \\ ReLuActivation \end{bmatrix} \times 2$ ↓ Add
Conv2	$[3 \times 3,\ 128] \times 2$	$[1 \times 1,\ 64] \times 1$ $[3 \times 3,\ 64] \times 1$ $[1 \times 1,\ 128] \times 1$	$[1 \times 1,\ 256] \times 1$ $[3 \times 3,\ 256] \times 1$ $[1 \times 1,\ 512] \times 1$	$[1 \times 1,\ 128] \times 1$ $[3 \times 3,\ 128] \times 1$ $[1 \times 1,\ 512] \times 1$		
Conv3	$[3 \times 3,\ 64] \times 2$	$[1 \times 1,\ 32] \times 1$ $[3 \times 3,\ 32] \times 1$ $[1 \times 1,\ 64] \times 1$	$[1 \times 1,\ 128] \times 1$ $[3 \times 3,\ 128] \times 1$ $[1 \times 1,\ 256] \times 1$	$[1 \times 1,\ 128] \times 1$ $[3 \times 3,\ 128] \times 1$ $[1 \times 1,\ 512] \times 1$		
Conv4	$[3 \times 3,\ 32] \times 2$	$[1 \times 1,\ 16] \times 1$ $[3 \times 3,\ 16] \times 1$ $[1 \times 1,\ 64] \times 1$	$[1 \times 1,\ 64] \times 1$ $[3 \times 3,\ 64] \times 1$ $[1 \times 1,\ 64] \times 1$	$[1 \times 1,\ 64] \times 1$ $[3 \times 3,\ 64] \times 1$ $[1 \times 1,\ 128] \times 1$		
Conv5	$[3 \times 3,\ 16] \times 2$	$[1 \times 1,\ 16] \times 1$ $[3 \times 3,\ 16] \times 1$ $[1 \times 1,\ 16] \times 1$	$[1 \times 1,\ 16] \times 1$ $[3 \times 3,\ 16] \times 1$ $[1 \times 1,\ 16] \times 1$	$[1 \times 1,\ 32] \times 1$ $[3 \times 3,\ 32] \times 1$ $[1 \times 1,\ 16] \times 1$		

Table 3. Hyper-parameters for segmentation model construction in this study.

Model	**Optimizer**	**Batch Size**	**Loss Function**	**Evaluation**
U-Net Link-Net	Adam	8	Binary cross-entropy	F1 score

2.5. Performance Evaluation

The performance of the segmentation models was evaluated with an F1 (Dice) score, which is a widely used indicator in semantic segmentation research. The F1 score can be calculated using precision and recall scores from the formulas in Table 4. TP, FP, TN, and FN denote the description in Table 4, and they can be calculated between the manually masked images and the predicted images from each model. All the metrics used within this study have been outlined in Table 4.

Table 4. Metrics description used within this study.

Metric	Description	Formula	Value Range	Unit
True Positive	Sum of correctly identified water pixels	TP	0~No. of pixels	ea
True Negative	Sum of correctly identified non-water pixels	TN	0~No. of pixels	ea
False Positive	Sum of pixels incorrectly identified as water	FP	0~No. of pixels	ea
False Negative	Sum of pixels incorrectly identified as non-water	FN	0~No. of pixels	ea
Precision (P)	Proportion of detected water pixels	$\frac{TP}{TP+FP}$	0~1	-
Recall (R)	Proportion of ground-truth water pixels detected	$\frac{TP}{TP+FN}$	0~1	-
F1 Score	Harmonic mean between the precision and recall	$2 \times \frac{R \times P}{R+P}$	0~1	-
Water level values	Observed water level at time i	y_i	0~water level	m
	Estimated water level at time i	$\hat{y}_i$	0~water level	m
	Average of observed water levels	$\overline{y}$	0~water level	m
R^2	Coefficient of determination	$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \overline{y})^2}$	0~1	-
MAE	Mean Absolute Error	$\frac{1}{n}\sum_{i=1}^{n}\lvert y_i - \hat{y}_i \rvert$	0~∞	m
RMSE	Root Mean Squared Error	$\sqrt{\frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{n}}$	0~∞	m
ME	Maximum Error	-	0~∞	m
$N_{E>0.05}$	Gross error (>0.05 m) number	-	0~313	ea

R^2 (Coefficient of Determination), MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), and ME (Maximum Error) metrics were used to evaluate the predicted water levels. The R^2 measures how well estimations are replicated by the model based on the proportion of total variation and unexplained variation. The MAE is used to quantify the size of the error to see how large the error is on average, and the RMSE has the characteristic of being sensitive to outliers. The ME represents the largest error and is a metric to check for robustness. In Table 4, y_i denotes the water levels measured from the ultrasonic sensor, $\overline{y}$ denotes the average water level, and $\hat{y}_i$ denotes the estimated water levels from this study.

3. Results and Discussion

3.1. Semantic Segmentation

3.1.1. Optimal Epoch Decision

This study compared the performances of two segmentation models, U-Net and Link-Net. In addition, four backbone models, ResNet-18, ResNet-50, VGGNet-16, and VGGNet-19, were applied to each segmentation model to compare performances by every epoch. Figure 3 shows the loss values obtained with four backbone models and U-Net by epoch 500. The dark solid line in Figure 3a indicates the validation losses of ReNet-50 and Figure 3b indicates those of VGGNet-19. The solid red line in Figure 3a indicates the validation losses of ResNet-18 and the solid red line in Figure 3b indicates those of VGGNet-16. Each dotted line indicates the training loss corresponding to its validation loss with the same color. In both graphs, the validation losses showed the smallest values around epoch 50, after which an overfitting problem occurred. Therefore, the U-Net model with datasets in this site showed optimal results around epoch 50. Comparing the performances of the backbone models, the ResNet series models outperformed the VGGNet series models in loss values and convergence speed. Also, in Figure 3a, ResNet-50 was more stable than ResNet-18 with less variation except for one point around epoch 50.

Figure 4 shows U-Net and Link-Net's training and validation losses with ResNet-50 as a backbone model. In the case of U-Net, overfitting occurred after epoch 50. In contrast, in the case of Link-Net, as the training loss became smaller, the validation loss did not converge to a certain value, making it difficult to select the optimal epoch before an occurrence of overfitting. Nevertheless, the lowest validation loss was around epoch 70 in Link-Net.

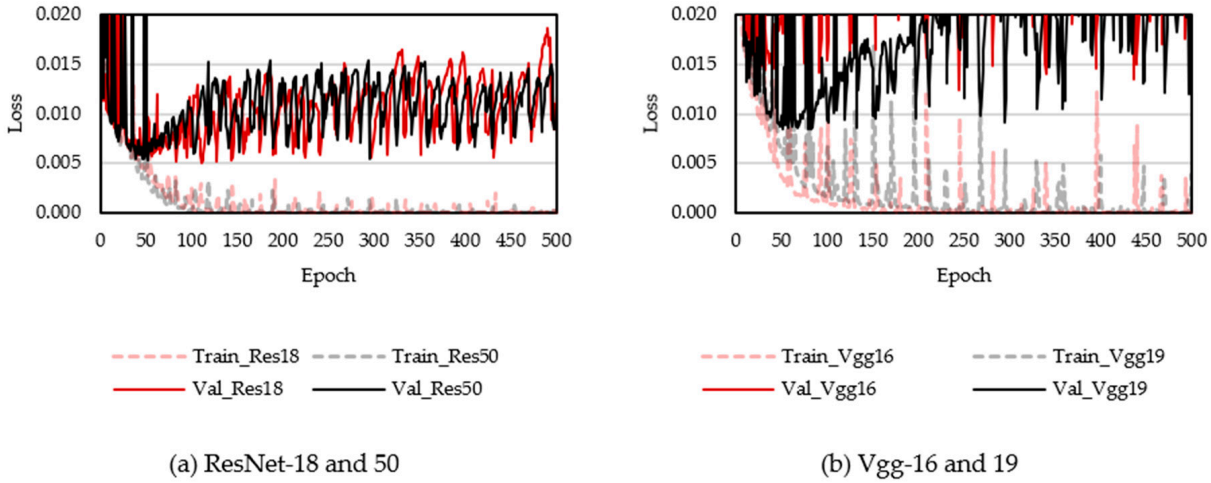


Figure 3. Results of backbone model comparison using ResNet-18, ResNet-50, VGGNet-16, and VGGNet-19 (U-Net segmentation): (**a**) train and validation losses with ResNet-18 and ResNet-50 for 500 epochs; (**b**) train and validation losses with VGGNet-16 and VGGNet-19 for 500 epochs.

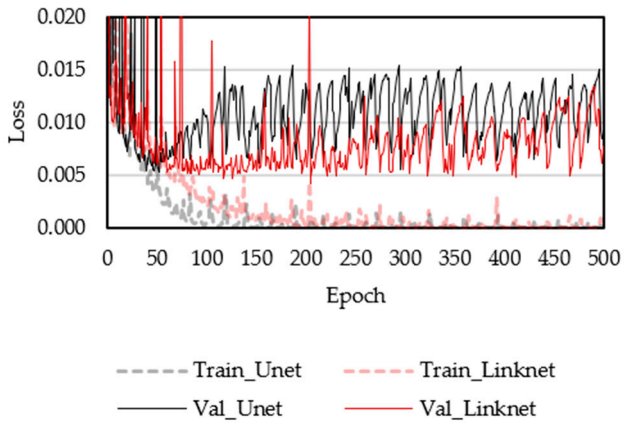


Figure 4. Segmentation model comparison using U-Net and Link-Net (ResNet-50 as backbone model): train and validation losses for 500 epochs.

Table 5 shows the number of parameters and training times for each combination to compute 500 epochs. As most parameters were located in the encoding path, training times mostly depended on the backbone model type. Link-Net did not show a significant advantage in terms of training speed over U-Net, which is due to the large size of the backbone model itself. For the VGGNet backbone models, the number of parameters was large, and there was no advantage in training time. Considering loss values by epochs and the training efficiency, ResNet outperformed VGGNet in backbone models.

Table 5. Number of parameters and training time for 500 epochs from each combination.

	Segmentation Model	**Backbone Model**			
		ResNet-18	**ResNet-50**	**VGGNet-16**	**VGGNet-19**
Number of parameters	U-Net	14,340,570	32,561,114	23,752,273	29,061,969
	Link-Net	11,521,690	28,783,386	20,325,137	25,634,833
Time to train (h)	U-Net	146	261	221	248
	Link-Net	144	256	226	252

Table 6 shows each combination's training and validation losses at the optimal epoch, where the validation loss was minimum. The validation losses of ResNet were lower than those of VGGNet, and there was no significant correlation with the segmentation model

type. To accurately compare the performance of each combination, an evaluation step using the test datasets was required. For each combination, the parameters of the epoch with the lowest validation loss were applied in the following section.

Table 6. Train and validation losses of each configuration and optimal epochs.

	Segmentation Model	Backbone Model			
		ResNet-18	**ResNet-50**	**VGGNet-16**	**VGGNet-19**
Train loss	U-Net	0.00242	0.00293	0.00238	0.00389
	Link-Net	0.00116	0.00357	0.00205	0.00270
Validation loss	U-Net	0.00517	0.00553	0.00866	0.00865
	Link-Net	0.00572	0.00521	0.00771	0.00969
Epoch	U-Net	57	55	57	53
	Link-Net	76	72	72	77

3.1.2. Segmented Results with 313 Test Datasets

Using the validation loss alone, it was difficult to accurately compare the performance of the combination with the ResNet-based backbone. Figure 5 shows the calculated F1 scores of 313 test datasets for the eight combinations. The input datasets were simplified by using ROI images, as in Figure 2. Although the performance of the generated mask does not absolutely follow the loss value, ResNet-based backbones outperformed VGGNet, and the difference between ResNet-18 and ResNet-50 was not significant. In addition, the difference between U-Net and Link-Net was far smaller than the difference among backbone models.

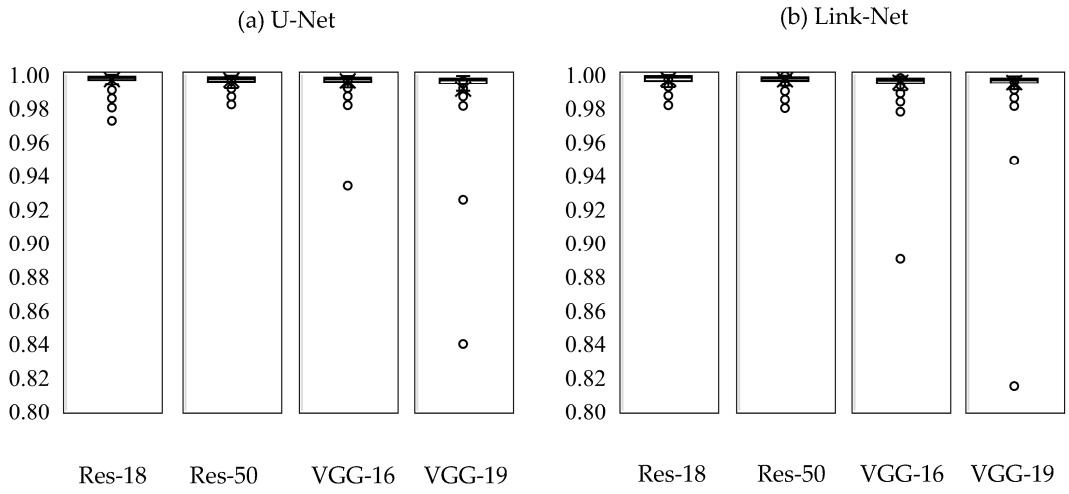


Figure 5. Box plots of the eight configuration results using 313 test datasets: (**a**) U-Net; (**b**) Link-Net.

Figure 6 presents the segmentation maps for four different combinations of full-resolution and ROI images. The three images from test datasets, which showed the lowest F1 scores, were selected to compare the performances of each configuration. For the first and third images, it was hard for human eyes to tell the water surface, but the U-Net series performed well on these images. The segmented mask images of U-Net showed outstanding results, especially in angular parts of the structure, while Link-Net segmented some parts as water pixels, which were unrelated to the water surface. Although the loss values of ResNet-50 in Table 6 were not the best, it performed well in terms of stability in Figure 4 and segmented results in Figure 6. Thus, this study applied ResNet-50 as the backbone model and U-Net as the segmentation model for water segmentation in this study.

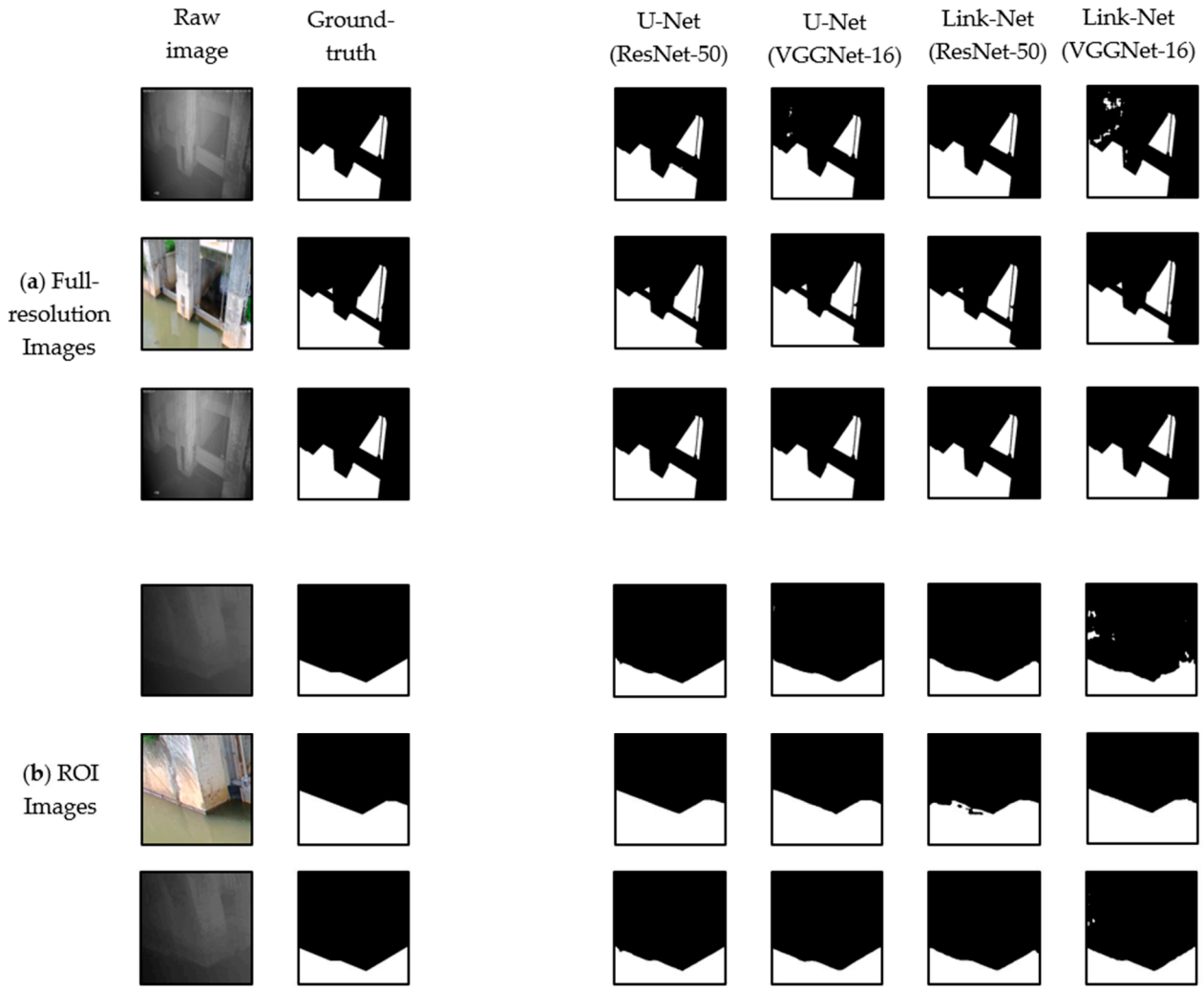


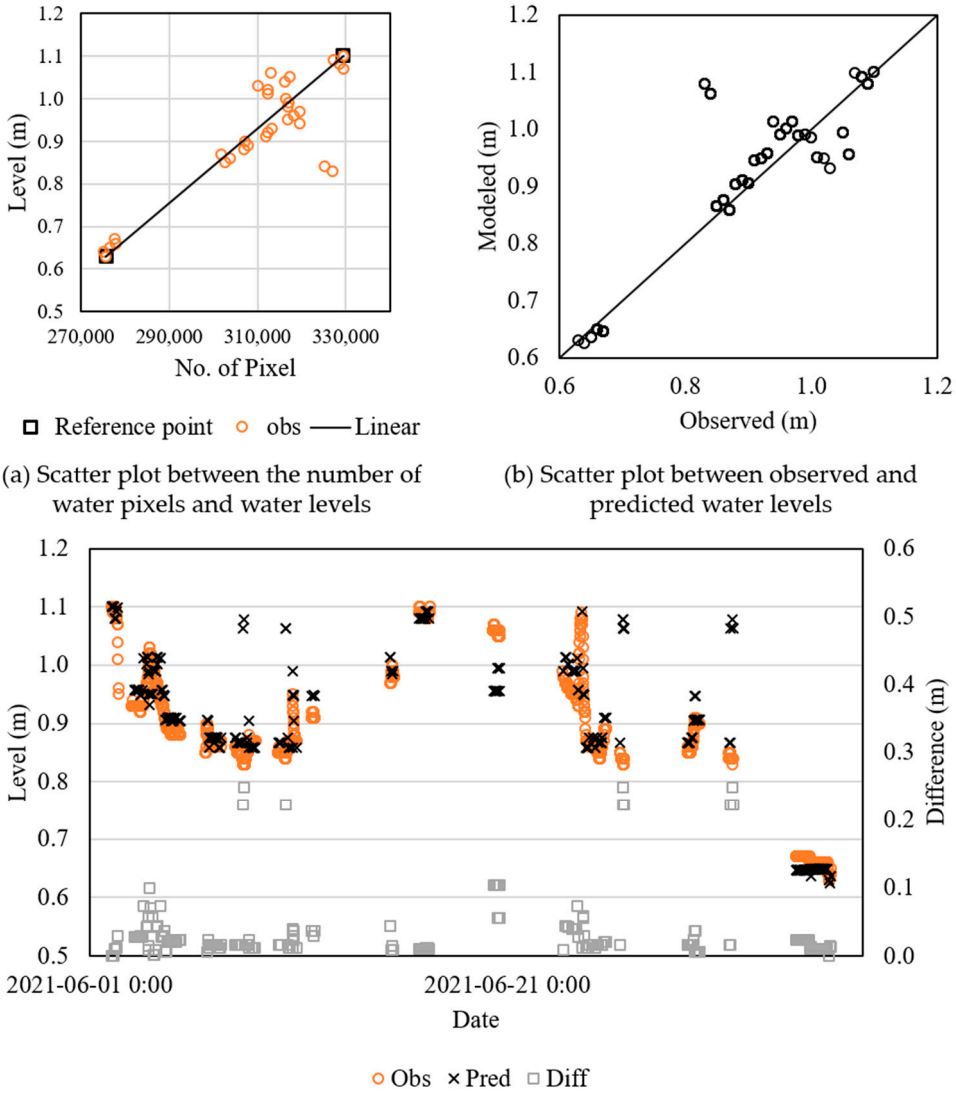
Figure 6. Comparison among different combinations: (**a**) The first three rows show the full-resolution images and (**b**) the other three rows show the ROI images. Original images and their associated ground-truth images are presented in the first two columns. Subsequent columns show the segmentation output of four combinations (U-Net or Link-Net and ResNet-50 or VGGNet-16).

3.2. Water Level Estimation

3.2.1. Full-Resolution Image and Linear Line for Conversion

Figure 7a shows the scatter plot between the number of water pixels and water levels in ground-truth mask images. Converting the water pixels into the water levels is necessary to estimate the water levels, and a linear line was derived using the number of pixels at the lowest and highest water levels. The R^2 of the linear line was calculated to be 0.76, and the time-series water level was calculated using the generated mask images and this linear line. Figure 7b shows the scatter plot between the predicted and observed water levels for 313 test datasets. The MAE was calculated to be 0.03 m, the RMSE to be 0.06 m, the ME to be 0.25 m, and the R^2 to be 0.84. Considering that the maximum water level in the channel was 1.10 m, the magnitude of the deviation was relatively large. Figure 7c shows the time-series data of the predicted and observed water levels. Obs, Pred, and Obs indicate the water levels measured using the ultrasonic sensor, the water levels simulated using a full-resolution image and a linear line, and the difference between Obs and Pred, respectively. In the time-series graph, the Obs shows all the values used for training, validation, and testing. As the test images were segmented almost identically to the ground-truth mask images with an F1 score of 0.997, the segmentation process was outstanding, whereas the process of converting water pixels into water levels performed poorly. This was because

the water pixels of the full-resolution image were not sufficiently correlated with the water level. Therefore, ROIs that were expected to have more parts correlated with the water level were selected from the full-resolution image.



(a) Scatter plot between the number of water pixels and water levels

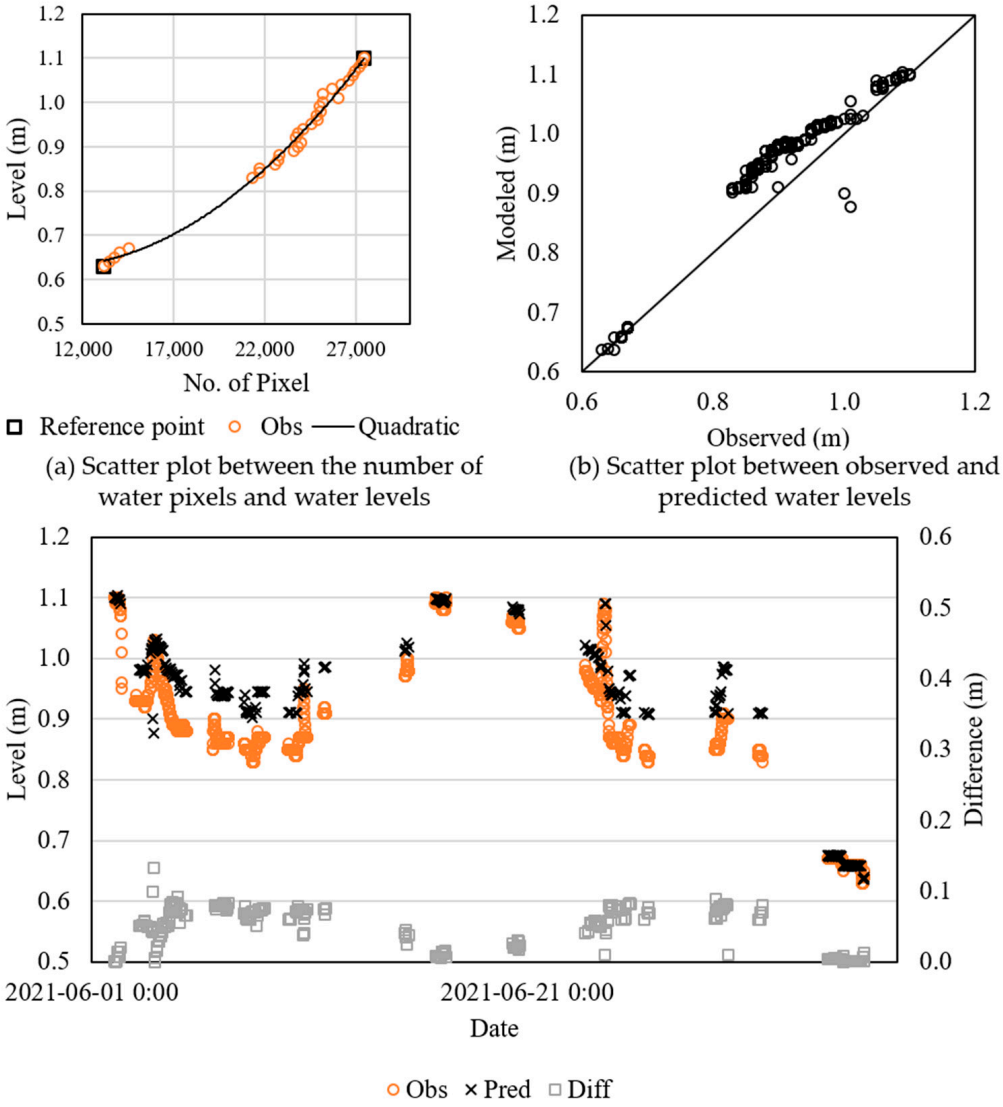
(b) Scatter plot between observed and predicted water levels

(c) Time-series water levels of observation, prediction, and difference results with randomly selected test data

Figure 7. Overall water level estimation results with full-resolution images and a linear line for conversion: (**a**) scatter plot between the number of water pixels and water levels; (**b**) scatter plot of observed and predicted water levels; and (**c**) time-series water levels of observation, prediction, and difference with randomly selected test data.

3.2.2. ROI Image and Linear Line for Conversion

The ROIs were selected, as shown in Figure 2, and cropped to a size of 256 × 256 × 3 from the full-resolution images. Figure 8a shows the scatter plot between water levels and water pixels from the ROI ground-truth mask images. As in Section 3.2.1, a linear line was derived using the number of pixels at the lowest and highest water levels from ROI mask images.



(a) Scatter plot between the number of water pixels and water levels

(b) Scatter plot between observed and predicted water levels

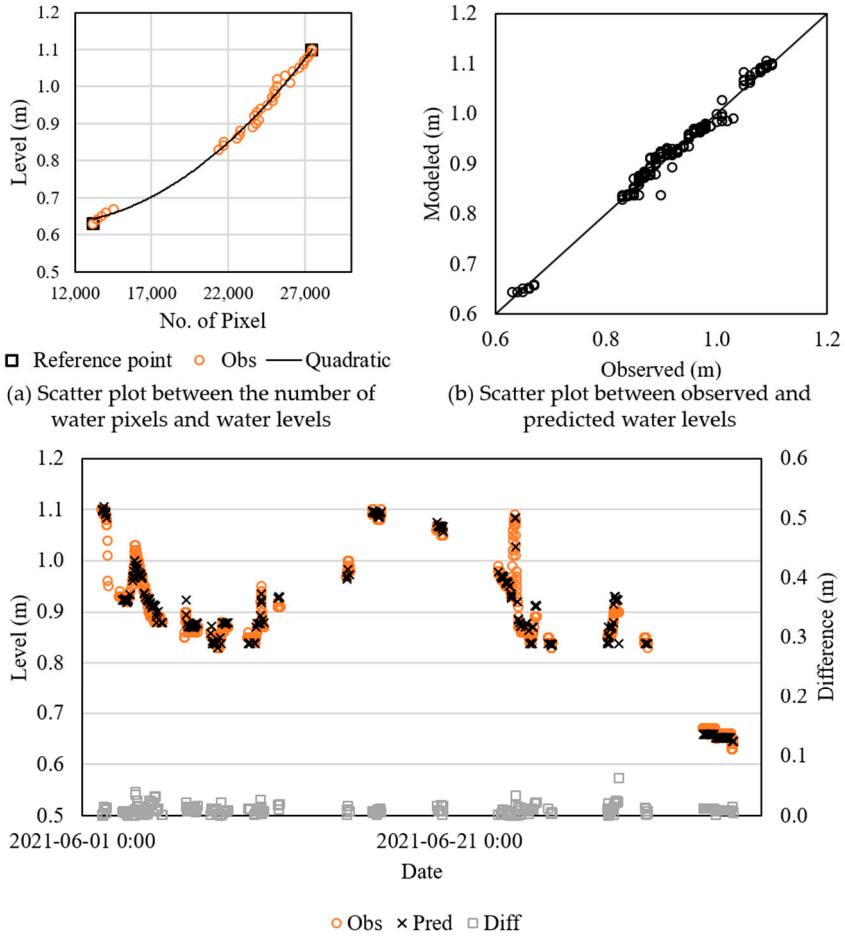
(c) Time-series water levels of observation, prediction, and difference results with randomly selected test data

Figure 8. Overall water level estimation results with ROI images and a linear line for conversion: (**a**) scatter plot between the number of water pixels and water levels; (**b**) scatter plot of observed and predicted water levels; and (**c**) time-series water levels of observation, prediction, and difference with randomly selected test data.

The R^2 of the linear line was calculated to be 0.94, and Figure 8b shows the scatter plot of the predicted and observed values for 313 test datasets. The MAE was calculated to be 0.05 m, the RMSE to be 0.06 m, the ME to be 0.13 m, and the R^2 to be 0.94. Compared to the case of not selecting an ROI, the metrics have improved, with deviations not exceeding 0.1 m for most of the test datasets. The time-series water levels were calculated as in Figure 8c. The test images were segmented almost identically to the ground-truth images with an F1 score of 0.999, which was higher compared to the case of not selecting ROIs. However, the water level deviations were still too large to apply, which was due to the deviations in the linear line in Figure 8b. Therefore, after selecting ROIs to segment images, this study applied a quadratic line for the pixel conversion instead of a linear line.

3.2.3. ROI Image and Quadratic Line for Conversion

This section also used the selected ROI image as in Section 3.2.2, and Figure 9a shows the scatter plot between water levels and water pixels from the ROI ground-truth mask images. The correlation between the water levels and pixels was curved rather than a linear line. Thereby, this section applied a quadratic line for the pixel conversion equation. The R^2 of the quadratic line was calculated to be 0.99, and Figure 9b shows the scatter plot of the observed and predicted values with ROI test images and a quadratic conversion line. The MAE was calculated to be 0.01 m, the RMSE to be 0.01 m, the ME to be 0.06 m, and the R^2 to be 0.99. The time-series water levels were calculated as in Figure 9c. In the graph, most of the predicted levels followed the observed levels closely. Compared to the previous cases, the metrics have improved significantly. In particular, the deviations did not exceed more than 0.05 m for most of the predicted levels, and the predicted water levels were highly correlated with the observed values.



(a) Scatter plot between the number of water pixels and water levels

(b) Scatter plot between observed and predicted water levels

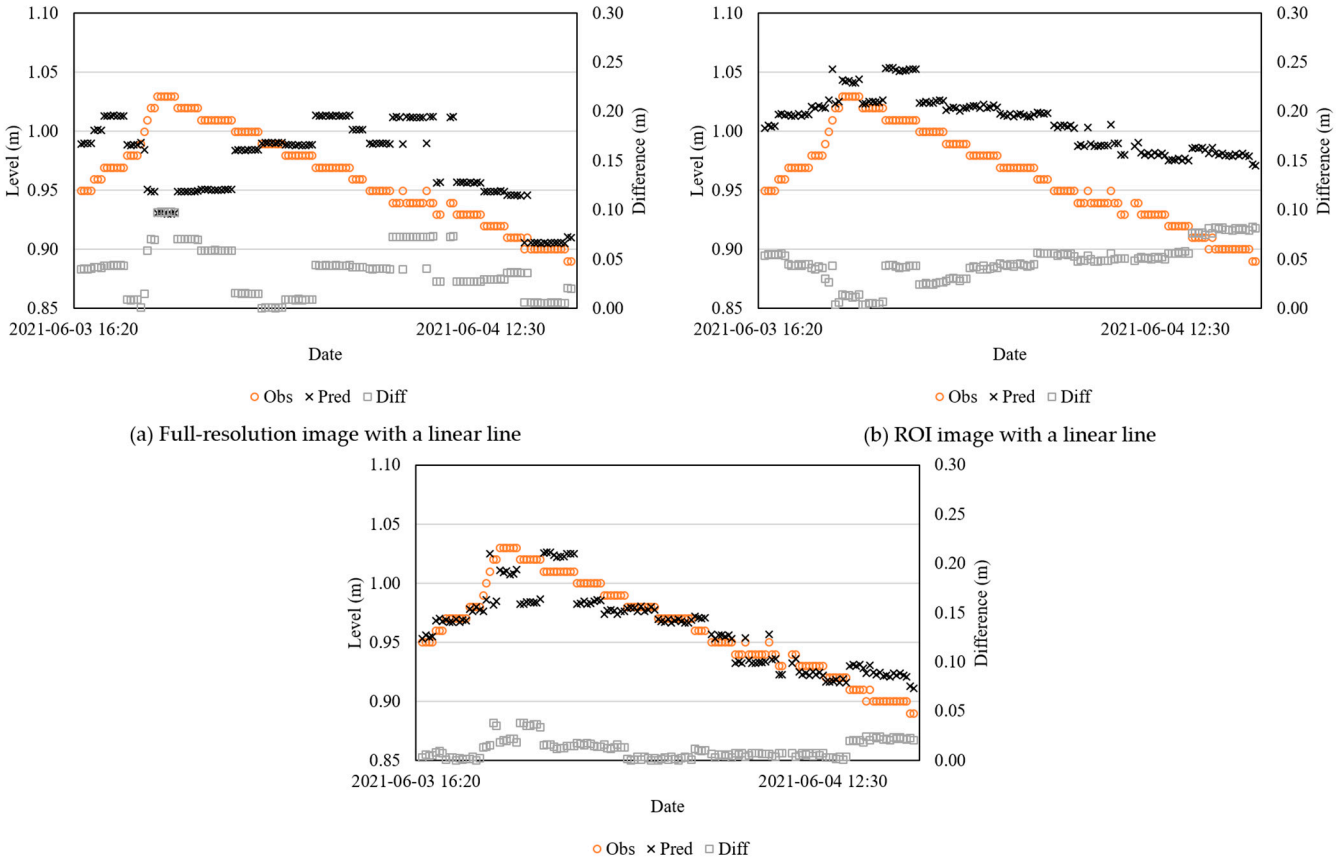
(c) Time-series water levels of observation, prediction, and difference results with randomly selected test data

Figure 9. Overall water level estimation results with ROI images and a quadratic line for conversion: (**a**) scatter plot between the number of water pixels and water levels; (**b**) scatter plot of observed and predicted water levels; and (**c**) time-series water levels of observation, prediction, and difference with randomly selected test data.

3.2.4. Overall Comparisons for Three Approaches

For data to be applied as a hydrological measuring device, it is important to acquire data with a constant time interval. This study selected a section of the dataset that provided continuous values at 10 min intervals with water level fluctuations. Figure 10 shows time-series water levels of observation, prediction, and difference with 10 min constant intervals.

Although the difference between the maximum and minimum levels in these datasets was smaller than in the previous graphs, analyzing the deviations in a narrower distribution is also important for the water level measurement. In Figure 10a,b, the overall deviations were large, and in Figure 10c, the largest deviations were seen around 0.90 and 1.03, only with a maximum of 0.04 m.



(a) Full-resolution image with a linear line

(b) ROI image with a linear line

(c) ROI image with a quadratic line

Figure 10. Time-series water levels of observation, prediction, and difference with constant 10 min interval from 3rd June 16:40 to 4th 17:00: (**a**) full-resolution image with a linear line, (**b**) ROI image with a linear line, and (**c**) ROI image with a quadratic line for conversion.

Table 7 shows the metric values for the water level estimation of the three cases with datasets selected randomly and with a constant 10 min interval. The MAE was calculated to be the largest for the ROI image with linear line, while the ME was largest for the full-resolution image with a linear line. This was due to the relatively small number of large deviations in full-resolution images with a linear line, whereas in ROI images with a linear line skewed values were constantly applied, affecting the average. This is also inferred from the gross error number ($N_{E>0.02}$ and $N_{E>0.03}$), and the ROI image with a quadratic line showed the highest metrics compared to the other cases. The datasets with a 10 min constant interval were selected for comparison and showed a maximum deviation of 0.04 m in the method of an ROI image with a quadratic line.

For the simulation of the dataset selected with a constant 10 min interval, the calculated MAE, RMSE, and ME were similar to the simulation of the randomly selected dataset. However, in Figure 10c, around 1.03 and 0.90 m sensor values, the predicted values showed low correlations, resulting in a smaller R^2. Since the segmentation showed high performances, the consideration of potential errors from the ultrasonic sensor is needed for the application of this methodology. Also, to improve the accuracy, it is necessary to verify the results with at least three different measurements, including staff gauges.

Table 7. Overall water level measurement results of three cases and two datasets in this study.

Dataset	Image and Conversion Line	R^2	MAE (m)	RMSE (m)	Maximum Error (m)	$N_{E>0.05}$	$N_{E>0.03}$	$N_{E>0.02}$	$N_{E>0.01}$
Dataset selected randomly	Full-resolution with linear	0.84	0.03	0.06	0.25	36	67	141	286
	ROI with linear	0.94	0.05	0.06	0.13	1	208	222	236
	ROI with quadratic	0.99	0.01	0.01	0.06	1	7	33	136
Dataset selected with constant 10 min interval	Full-resolution with linear	0.05	0.04	0.05	0.10	39	83	102	111
	ROI with linear	0.86	0.05	0.05	0.08	59	114	129	135
	ROI with quadratic	0.86	0.01	0.01	0.04	0	9	29	65

4. Discussion

The CCTV cameras installed in irrigation channels are mostly exposed to the outside environment. Therefore, it is necessary to provide sufficient results even in harsh environments during the irrigation season. The example images in Figure 6 show the heaviest fog at night and the strongest sunlight during the day. When the fog was heavy, it was hard to distinguish the water surface with human eyes, while the U-Net model with ResNet-50 was able to segment the water surface similarly to the ground-truth image. Also, the F1 score of the segmented image was as high as the results of other test datasets. Nevertheless, the possibility of more severe fog cannot be ruled out, and there were not enough images representing heavy rainfall, which should be considered for the application. In addition, there is a possible problem that the camera oscillates so excessively due to strong winds or rainfall that the viewpoints are not completely the same. In the datasets collected in this study, the maximum deviation vertically was 3 pixels, and the maximum deviation horizontally was 4 pixels. Considering that the resolution of the ROI images was 256×256, the difference between the affected and the unaffected was up to 1780 pixels out of 65,536 pixels. Although further research is needed on the effects and solutions in more harsh environments, it seemed that errors were within the acceptable range for the datasets in this study.

From the viewpoint of the channel in this study, the full-resolution images contained a lot of water surfaces unrelated to the water level, which led to the low correlation between the water pixels and the observed water levels. Therefore, this study selected an ROI in the image most correlated with the water level. Concerning the generalization of this methodology, there are two possible approaches: one is to train the model on the datasets of all the target channels, and the other is to train separate models for each target channel. The optimal performance was seen around epoch 50, and it took about 15 h of training with the hardware used in this study. Generalization would require more time to train and better-performing hardware. However, compared to other research fields of applying semantic segmentation that segments various objects from complex backgrounds, this methodology is expected to show sufficient results with less time and labor and can be easily applied without major modifications once built. Furthermore, if data augmentation technology is applied in the dataset construction process, the labor required to construct the datasets is expected to decrease. Nevertheless, there is a disadvantage in that ROIs must be selected on each site, which is laborious and uncertain.

Concerning the water level measurement of this study, it was analyzed with the allowable errors in the irrigation system and with the research results related to the water level measurement using images and sensors. Costabile [3] showed an RMSE of 2.7 to 6.9 cm between simulation and sensor values in a border irrigation 2D simulation, but it is difficult to make a simple comparison as the measurements were taken in the field rather

than in the channel. Liu [17] applied filtering and binarization processes to CCTV camera images in the irrigation and drainage channel to estimate the water level by obtaining the number of pixels in the vertical direction and compared it with the radar wave level sensor, reporting an RMSE of 0.05 to 0.18 m and MAE of 0.04 to 0.20 m. Though the image processing method and the size of the channel were different, this study showed smaller error indices.

In research on applying semantic segmentation techniques to estimate urban flood water depth, refs. [37,38] reported an MAE of 0.10 m and RMSE of 0.11 m using social media images. In studies applying a staff gauge, Muhadi [21] reported $N_{E>0.01}$ in 46 out of 1152 trials, and Akbari [2] reported an RMSE of 0.01 m. Previous studies have shown that higher accuracy is expected if the staff gauge is applied from a fixed viewpoint. The application at the irrigation channel would be easier and more accurate if a staff gauge was installed.

The water loss rate at the irrigation channel varies depending on multiple factors. These include whether the channel is main or distributary, whether it is lined or not, climate conditions, the economic status of the country, and the shape or condition of the structure [39,40]. However, irrigation channels are generally known to have a loss of 5 to 10% for a concrete-lined channel and 10 to 25% for an earthen channel [39,41]. Likewise, when estimating the quantities of the water supply, there is a margin of allowable errors. Also, concerning the decision making of the irrigation system, the accuracy required from the measurement is relatively low. Although additional applications are needed based on the datasets from other sites and future research is needed to improve the measurement results, this methodology could be an accurate measuring approach with low-cost equipment.

5. Conclusions

In this study, images obtained from a CCTV camera were used to measure water levels in an irrigation channel. The images were captured with the same viewpoint at a 10 min interval, and the water levels measured from an ultrasonic sensor simultaneously were used as observed values. Images were combined with manually annotated mask images to form datasets, and datasets were used for constructing the segmentation models. From the comparison analysis, the U-Net with a ResNet-50 backbone model showed the most stable and well-segmented mask images. The water levels were then predicted using the water pixels in the generated mask images, and after ROIs were selected the quadratic line showed acceptable performances for the application.

Measuring water levels in irrigation channels is an important task in terms of irrigation system decision making and estimating the quantity of irrigation water supplies. It is necessary to consider the purpose and role of the irrigation channel, as well as the cost and accuracy of the measurement. Most previous research measuring water levels in irrigation channels or paddy fields applied the ultrasonic sensor, which is a costly inefficient device for irrigation purpose. This study proposes a method for measuring water levels in irrigation channels based on the acceptable accuracy and cost for the application.

The results of this study can be used as reference data for decision making in irrigation systems and as an auxiliary device for estimating the quantity of irrigation water supplies by compensating for the possible data loss from other measurements. The next goal is to generalize this methodology in other irrigation channels and overcome more harsh environments, such as wind, heavy rain, and severe fog. The challenge will be to increase the accuracy and improve the efficiency of constructing the datasets in the more severe environments expected to occur. Moreover, it is also important to devise an objective and robust methodology for selecting ROIs in other channels.

Author Contributions: Methodology, K.K.; Software, K.K.; Validation, K.K.; Formal analysis, K.K.; Writing—original draft, K.K.; Writing—review & editing, J.-Y.C.; Supervision, J.-Y.C.; Project administration, J.-Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in the current study are available from the corresponding author upon reasonable request.

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

References

1. Koech, R.; Langat, P. Improving irrigation water use efficiency: A review of advances, challenges and opportunities in the Australian context. *Water* **2018**, *10*, 1771. [CrossRef]
2. Akbari, M.; Gheysari, M.; Mostafazadeh-Fard, B.; Shayannejad, M. Surface irrigation simulation-optimization model based on meta-heuristic algorithms. *Agric. Water Manag.* **2018**, *201*, 46–57. [CrossRef]
3. Costabile, P.; Costanzo, C.; Gangi, F.; De Gaetani, C.I.; Rossi, L.; Gandolfi, C.; Masseroni, D. High-resolution 2D modelling for simulating and improving the management of border irrigation. *Agric. Water Manag.* **2023**, *275*, 108042. [CrossRef]
4. Conde, G.; Quijano, N.; Ocampo-Martinez, C. Modeling and control in open-channel irrigation systems: A review. *Annu. Rev. Control* **2021**, *51*, 153–171. [CrossRef]
5. Koech, R.; Rod, S.; Malcolm, G. Automation and control in surface irrigation systems: Current status and expected future trends. In Proceedings of the Southern Region Engineering Conference, Toowoomba, Australia, 11–12 November 2010.
6. Weyer, E. Control of irrigation channels. *IEEE Trans. Control Syst. Technol.* **2008**, *16*, 664–675. [CrossRef]
7. Lee, J. Evaluation of automatic irrigation system for rice cultivation and sustainable agriculture water management. *Sustainability* **2022**, *14*, 11044. [CrossRef]
8. Kuswidiyanto, L.W.; Nugroho, A.P.; Jati, A.W.; Wismoyo, G.W.; Murtiningrum; Arif, S.S. Automatic water level monitoring system based on computer vision technology for supportin the irrigation modernization. *IOP Conf. Ser. Earth Environ. Sci.* **2021**, *686*, 012055. [CrossRef]
9. Lozano, D.; Arranja, C.; Rijo, M.; Mateos, L. Simulation of automatic control of an irrigation canal. *Agric. Water Manag.* **2010**, *97*, 91–100. [CrossRef]
10. Masseroni, D.; Moller, P.; Tyrell, R.; Romani, M.; Lasagna, A.; Sali, G.; Facchi, A.; Gandolfia, C. Evaluating performances of the first automatic system for paddy irrigation in Europe. *Agric. Water Manag.* **2018**, *201*, 58–69. [CrossRef]
11. Hamdi, M.; Rehman, A.; Alghamdi, A.; Nizamani, M.A.; Missen, M.M.S.; Memon, M.A. Internet of Things (IoT) Based Water Irrigation System. *Int. J. Online Biomed. Eng.* **2021**, *17*, 69–80. [CrossRef]
12. Park, C.E.; Kim, J.T.; Oh, S.T. Analysis of stage-discharge relationships in the irrigation canal with auto-measuring system. *J. Korean Soc. Agric. Eng.* **2012**, *54*, 109–114.
13. Hong, E.M.; Nam, W.-H.; Choi, J.-Y.; Kim, J.-T. Evalutation of water supply adequacy using real-time water level monitoring system in paddy irrigation canals. *J. Korean Soc. Agric. Eng.* **2014**, *56*, 1–8.
14. Lee, J.; Noh, J.; Kang, M.; Shin, H. Evaluation of the irrigation water supply of agricultural reservoir based on measurement information from irrigation canal. *J. Korean Soc. Agric. Eng.* **2020**, *62*, 63–72.
15. Seibert, J.; Strobi, 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]
16. Kuo, L.-C.; Tai, C.-C. Automatic water-level measurement system for confined-space applications. *Rev. Sci. Instrum.* **2021**, *92*, 085001. [CrossRef]
17. Liu, W.-C.; Chung, C.-K.; Huang, W.-C. Image-based recognition and processing system for monitoring water levels in an irrigation and drainage channel. *Paddy Water Environ.* **2023**, *21*, 417–431. [CrossRef]
18. Sharma, N.; Sheifali, G.; Deepika, K.; Sultan, A.; Hani, A.; Yousef, A.; Asadullah, S. U-Net model with transfer learning model as a backbone for segmentation of Gastrointestinal tract. *Bioengineering* **2023**, *10*, 119. [CrossRef]
19. Long, J.; Evan, S.; Trevor, D. Fully convolutional networks for semantic segmentation. *arXiv* **2015**, arXiv:1411.4038.
20. Zhou, L.; Zhang, C.; Wu, M. D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018.
21. Muhadi, N.A.; Abdullah, A.F.; Bejo, S.K.; Mahadi, M.R.; Mijic, A. Deep Learning semantic segmentation for water level estimation using surveillance camera. *Appl. Sci.* **2021**, *11*, 9691. [CrossRef]
22. Vianna, P.; Farias, R.; de Albuquerque Pereira, W.C. U-Net and SegNet performances on lesion segmentation of breast ultrasonography images. *Res. Biomed. Eng.* **2021**, *37*, 171–179. [CrossRef]
23. Chang, R.; Hou, D.; Chen, Z.; Chen, L. Automatic extraction of urban impervious surface based on SAH-Unet. *Remote Sens.* **2023**, *15*, 1042. [CrossRef]
24. Abdollahi, A.; Pradhan, B.; Alamri, A.M. An ensemble architecture of deep convolutional Segnet and Unet networks for building semantic segmentation from high-resolution aerial images. *Geocarto Int.* **2022**, *37*, 3355–3370. [CrossRef]
25. Kim, J.; Jeon, H.; Kim, D.-J. Extracting flooded areas in Southeast Asia using SegNet and U-Net. *Korean J. Remote Sens.* **2020**, *36*, 1095–1107.
26. Hies, T.; Parasuraman, S.B.; Wang, Y.; Duester, R.; Eikaas, H.S.; Tan, K.M. Enhanced water-level detection by image processing. In Proceedings of the 10th International Conference on Hydroinformatics, Hamburg, Germany, 14–18 July 2012.

27. Lin, Y.-T.; Lin, Y.-C.; Han, J.-Y. Automatic water-level detection using single-camera images with varied poses. *Measurement* **2018**, *127*, 167–174. [CrossRef]
28. De Vitry, M.M.; Kramer, S.; Wegner, J.D.; Leitão, J.P. Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network. *Hydrol. Earth Syst. Sci.* **2019**, *23*, 4621–4634. [CrossRef]
29. Zaffaroni, M.; Rossi, C. Water segmentation with deep learning models for flood detection and monitoring. In Proceedings of the 17th Information Systems for Crisis Response and Management, Blacksburg, VA, USA, 24–27 May 2020.
30. Lopez-Fuentes, L.; Rossi, C.; Skinnemoen, H. River segmentation for flood monitoring. In Proceedings of the 2017 IEEE International Conference on Big Data, Boston, MA, USA, 11–14 December 2017.
31. Akiyama, T.S.; Junior, J.M.; Gonçalves, W.N.; Bressan, P.O.; Eltner, A.; Binder, F.; Singer, T. Deep learning applied to water segmentation. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci. (ISPRS Arch.)* **2020**, *43*, 1189–1193. [CrossRef]
32. Vandaele, R.; Dance, S.L.; Ojha, V. Deep learning for automated river-level monitoring through river-camera images: An approach based on water segmentation and transfer learning. *Hydrol. Earth Syst. Sci.* **2021**, *25*, 4435–4453. [CrossRef]
33. Bai, G.; Hou, J.; Zhang, T.; Li, B.; Han, H.; Wang, T.; Hinkelmann, R.; Zhang, D.; Guo, L. An intelligent water level monitoring method based on SSD algorithm. *Measurement* **2021**, *185*, 110047. [CrossRef]
34. Kim, K.; Kim, M.; Yoon, P.; Bang, J.; Myoung, W.-H.; Choi, J.-Y.; Choi, G.-H. Application of CCTV images and semantic segmentation model for water level estimation of irrigation channel. *J. Korean Soc. Agric. Eng.* **2022**, *64*, 63–73.
35. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. *arXiv* **2015**, arXiv:1505.04597v1.
36. Chaurasia, A.; Culurciello, E. LinkNet: Exploiting encoder representations for efficient semantic segmentation. *arXiv* **2017**, arXiv:1707.03718v1.
37. Chaudhary, P.; D'Aronco, S.; Leitão, J.P.; Schindler, K.; Wegner, J.D. Water level prediction from social media images with a multi-task ranking approach. *ISPRS J. Photogramm. Remote Sens.* **2020**, *167*, 252–262. [CrossRef]
38. Zhang, Z.; Zhou, Y.; Liu, H.; Gao, H. In-situ water level measurement using NIR-imaging video camera. *Flow Meas. Instrum.* **2019**, *67*, 95–106. [CrossRef]
39. Lee, Y.-J.; Kim, P.-S.; Kim, S.J.; Jee, Y.K.; Joo, U.J. Estimation of water loss in irrigation canals through field measurement. *J. Korean Soc. Agric. Eng.* **2008**, *50*, 13–21.
40. Mohammadi, A.; Rizi, A.P.; Abbasi, N. Field measurement and analysis of water losses at the main and tertiary levels of irrigation canals: Varamin Irrigation Scheme, Iran. *Glob. Ecol. Conserv.* **2019**, *18*, e00646. [CrossRef]
41. Sultan, T.; Latif, A.; Shakir, A.S.; Kheder, K.; Rashid, M.U. Comparison of water conveyance losses in unlined and lined watercourses in developing countries. *Tech. J. Univ. Eng. Technol. Taxila* **2014**, *19*, 23.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.