Article

Deep Learning Semantic Segmentation for Water Level Estimation Using Surveillance Camera

Nur Atirah Muhadi ^{1,*} , Ahmad Fikri Abdullah ^{1,2,*}, Siti Khairunniza Bejo ¹ , Muhammad Razif Mahadi ¹ and Ana Mijic ³

¹ Department of Biological and Agricultural Engineering, Faculty of Engineering, Universiti Putra Malaysia, Serdang 43400, Malaysia; skbejo@upm.edu.my (S.K.B.); razifman@upm.edu.my (M.R.M.)

² Institute of Aquaculture and Aquatic Sciences, Batu 7, Jalan Kemang 6, Teluk Kemang, Port Dickson 70150, Malaysia

³ Department of Civil and Environmental Engineering, South Kensington Campus, Imperial College London, London SW7 2AZ, UK; ana.mijic@imperial.ac.uk

* Correspondence: gs53332@student.upm.edu.my (N.A.M.); ahmadfikri@upm.edu.my (A.F.A.); Tel.: +603-97694337 (A.F.A.)

Citation: 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. <https://doi.org/10.3390/app11209691>

Academic Editor: Yosoon Choi

Received: 20 August 2021

Accepted: 10 September 2021

Published: 18 October 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 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/>).

Abstract: The interest in visual-based surveillance systems, especially in natural disaster applications, such as flood detection and monitoring, has increased due to the blooming of surveillance technology. In this work, semantic segmentation based on convolutional neural networks (CNN) was proposed to identify water regions from the surveillance images. This work presented two well-established deep learning algorithms, DeepLabv3+ and SegNet networks, and evaluated their performances using several evaluation metrics. Overall, both networks attained high accuracy when compared to the measurement data but the DeepLabv3+ network performed better than the SegNet network, achieving over 90% for overall accuracy and IoU metrics, and around 80% for boundary F1 score (BF score), respectively. When predicting new images using both trained networks, the results show that both networks successfully distinguished water regions from the background but the outputs from DeepLabv3+ were more accurate than the results from the SegNet network. Therefore, the DeepLabv3+ network was used for practical application using a set of images captured at five consecutive days in the study area. The segmentation result and water level markers extracted from light detection and ranging (LiDAR) data were overlaid to estimate river water levels and observe the water fluctuation. River water levels were predicted based on the elevation from the predefined markers. The proposed water level framework was evaluated according to Spearman's rank-order correlation coefficient. The correlation coefficient was 0.91, which indicates a strong relationship between the estimated water level and observed water level. Based on these findings, it can be concluded that the proposed approach has high potential as an alternative monitoring system that offers water region information and water level estimation for flood management and related activities.

Keywords: flood detection; deep learning; water level estimation; water segmentation; CCTV; CNN

1. Introduction

The frequency and impact of flood disasters have continued to escalate, causing losses of human life and damages to property due to extreme changes in climate events. The effects of extreme weather have been observed in many countries. For instance, in the UK, a series of flooding events occurred in late 2013 and early 2014, caused by continuous major storms hitting the UK, eliciting a major emergency response and media exposure [1]. Malaysia also experienced an increase in rainfall extreme in December 2014, which led to the worst flooding in decades [2], affecting more than 200,000 residents and more than RM2.5 billion economic losses [3]. In recent years, Brazil experienced heavy rainfall, setting new rainfall records at certain locations [4]. For instance, Santo Agostinho had the wettest

period in 110 years while Sao Paulo experienced the wettest period in 77 years in January 2020, causing 100 deaths.

There were 367 disaster events on average per year worldwide, with floods accounting for 44% of total events [5]. The flood issue has gained attention globally with significant efforts being made to develop effective flood prevention and monitoring solutions. Currently, the primary approaches to monitoring flood events based on data sources used are sensor measurements and remote sensing [6–11]. The common flood monitoring system obtains data from gauge stations such as water level stations and rainfall stations [12,13]. The on-site stations use sensors that allow real-time measurement of water levels or rainfall levels. The data are then used to forecast flood events by developing dynamic flood modeling that provides advanced flood information for flood management and mitigation. However, the main restriction of using these sensors is the limited number of sensor installations due to the high cost of installation and maintenance [14]. Besides, the water level sensor provides only point-based information and the data cannot be visualized, thus it does not accurately represent the actual condition at sites.

Remote sensing technology is another option to monitor flood events instead of direct sensor measurements. Optical imagery and radar imagery are usually adopted for monitoring floods within a large-scale area that can cover the entire flood events [15–19]. However, it is not easy to extract information within a specific small area from satellite imagery. Besides, satellite images are normally affected by vegetation canopies and cloud covers which require post-processing work before the data can be used. Time delay due to complex processing work and satellite orbital cycles are other restrictions pertaining to the use of satellite imagery in flood applications. On the other hand, unmanned aerial vehicles (UAV), another option in remote sensing, can provide more detailed information in specific, small or large flood areas [20–22]. UAVs could be operated autonomously using a pre-programmed flight plan, which is a wise choice during flood events. However, when it is being operated during poor weather conditions, especially during heavy rain and windy conditions, the performance of a UAV could be affected in terms of data accuracy and resolution [23–26].

Due to the limitations in current methods used for flood monitoring, the surveillance camera has become a popular alternative to be used as an input source for flood monitoring events, especially in a small-scale areas [14,27]. The information obtained from the surveillance camera could be in still images [28,29] or video footage [30,31] and could be extracted using various computer vision techniques. Computer vision is the common approach to capture and process the data from a surveillance camera for flood disaster applications by employing a visual sensor [32,33]. For instance, image segmentation is the common computer algorithm used to extract useful information from digital images. It classifies the image into several regions according to the attributes of its pixels such as color or texture information [34]. Over the last decade, conventional image segmentation methods, such as thresholding, region-based, or hybrid segmentation, were often applied to segment water regions from the background [27,35,36]. For example, Lo et al. [37] proposed a graph-based segmentation method where a virtual seed is used as guidance for flood region detection from images captured by the surveillance camera. Even though the graph-based and other segmentation methods proved to be capable of extracting flood information from the image, it is found that the methods are often handcrafted for specific data as prior knowledge of the data needs to be analyzed [28]. Hence, it may be difficult to apply these methods to other data images. Therefore, the segmentation task needs a state-of-the-art approach that can be adopted for all types of data complexity and can be applied to a large number of surveillance cameras.

In recent years, deep learning approach has gained attention from researchers all over the world. It has been extensively applied in digital voice assistants, such as Siri, Google Assistant, and Alexa, as well as in self-driving cars. Deep learning is the key technology behind the autonomous vehicles that teaches computers to learn by example. Autonomous cars use deep learning semantic segmentation techniques to identify vehicles,

pedestrians, traffic signs, and other road features. Hence, it has brought great changes and improvements in life with many impossible tasks now achievable using deep learning. With the evolvement of convolutional neural networks (CNN), many researchers have started applying this method in their studies as it can deal with large amounts of training data and achieve high accuracy results. However, since CNN models are data-hungry, a huge amount of data and computational power are needed. Thus, transfer learning is often used to address the issue of data scarcity and to improve the target task performance with less training time by using the pre-trained network. Transfer learning uses a pre-trained network as a starting point to learn a new task. Fine-tuning the pre-trained network is much easier and faster than training with randomly initialized weights from scratch. It is commonly used in deep learning applications, especially when dealing with a small number of training images. Past studies have proposed a variety of deep learning architectures for various tasks such as image classification [38–40], object detection [41–43], as well as semantic segmentation [44–47]. In the field of flood applications, one of the early studies that used CNN to identify flooding areas was that of Bischke et al. [48]. The authors retrieved flooded regions from satellite images by using a segmentation network. A similar study was carried out by Yu et al. [49], who used CNN to extract water bodies from multispectral data. The study used CNN and then a logistic classifier for water body extraction and compared the results with an artificial neural network (ANN) and support vector machine (SVM). The findings show that the proposed method achieved higher accuracies compared to ANN and SVM.

In addition, a study on disaster detection from aerial imagery using CNN was done by Amit and Aoki [50]. In this study, CNN was implemented in detecting the changes between pre-disaster and post-disaster imagery for both landslide and flood disasters. The proposed method obtained results between 80% and 90% of the f-score, which proved that CNN can detect a disaster region at relatively high accuracy. A similar study was carried out by Isikdogan et al. [51] that adopted the CNN approach to map surface water on Landsat imagery that could be used in predicting flood disasters. The trained model has successfully segmented water from land, snow, ice, clouds, and shadows and the model could be used in classifying other types of land cover that involve satellite imagery. In addition, Chen et al. [52], Feng et al. [53], and Li et al. [54] performed similar studies that applied deep learning for water body extraction from high-resolution satellite imagery. Even though they used different CNN architectures, all findings show that deep learning performed well as compared to conventional approaches.

Despite many deep learning studies that have been performed on remote sensing imagery in flood applications, there are only a few studies that used the deep learning approaches, in particular water segmentation methods, to exploit digital images, especially from the surveillance camera. A study carried out by Lopez-Fuentez et al. [29] used a deep learning technique to extract flooded regions from surveillance camera images. This work performed several algorithms for water segmentation, namely FCN-8s, DenseNet, and Pix2Pix, to monitor river water levels during flood events to determine the efficiency of each algorithm. Next, Moy de Vitry et al. [55] applied a deep convolutional neural network to detect floodwater and introduced a static observer flooding index (SOFI) to determine water level fluctuations. Water segmentation in this work was developed based on U-net architecture and obtained IoU higher than 90% on average. Successful work has been done by Akiyama et al. [56] that applied CNN based on SegNet network to segment river water from camera images. The authors installed the camera in one location and then captured images for 50 consecutive days to represent different conditions and ambient. The images were then labeled and used as the input dataset for the network. The performance of SegNet achieved around 97% and 98% for pixel accuracy and IoU metrics, respectively. Regardless of the high performance, the author did not apply the trained network on images from different locations, which is why the results from this study were not included for comparison. Vandaele et al. [57] investigated the use of deep learning for water levels estimation from the surveillance images. The authors utilized UperNet network and

DeepLabV3 network for the water segmentation process and reported that DeepLabV3 obtained better performance than the UperNet network. Then, water levels were estimated using landmark-based water-level estimation (LBWLE) and SOFI index [55].

Based on the previous studies conducted, it can be concluded that Moy de Vitry et al. [55] and Vandaele et al. [57] only focused on water level fluctuation and estimation, whereas this paper highlights water level thresholds that consist of normal, alert, warning and danger levels for flood alert. In particular, this paper explores the use of transfer learning approaches for image segmentation, in the context of flood detection. In this work, two powerful semantic segmentation networks, DeepLabv3+ and SegNet, have been adopted to extract the water body regions from the surveillance images. DeepLabv3+ network was created with weights initialized from a pre-trained Resnet-18 network, while the SegNet model adapted the VGG-16 network [58] for segmenting the water bodies. The networks were trained, tested, and evaluated to determine their performance. During the inference phase, new images from several locations were used on the networks to segment the water regions from the background to find out how widely applicable the networks are. The network that shows better performance was then used to segment a series of images captured for five consecutive days in one specific location. The segmentation results were extracted and post-processing steps took place to enhance the water features. Next, water level information generated from light detection and ranging (LiDAR) data were overlaid with the post-processed image to estimate the river water level, determine the threshold levels, and observe the fluctuation of water level in the study area visually.

2. Methodology

2.1. Data Collection and Preparation

This paper relied on existing flood-related images from the European Flood 2013 Dataset (<http://www.inf-cv.uni-jena.de/Research/Datasets/European+Flood+2013.html> (accessed on 15 December 2020)) as well as local images captured from surveillance cameras installed by the Department of Irrigation and Drainage (DID), Malaysia. The European Flood 2013 dataset was developed by the Computer Vision Group of the University of Jena. Most of the images are related to the central European floods that occurred in May and June 2013 and have been collected in July 2017 from the Wikimedia Commons Category “Central Europe floods, May/June 2013” and its sub-categories that show flooding images. To ensure the data used in this experiment are reliable, each photo was evaluated manually and only 600 images were used from the complete set of 3710 images. Images that did not meet the criteria, such as blurred photos, non-flood-related photos, and water that was not representative of flooding events such as polluted water, were excluded. In addition, images captured from surveillance cameras collected from 20 different stations in Selangor that were retrieved from the official website of DID Selangor (<http://infobanjirjps.selangor.gov.my> (accessed on 15 December 2020)), were also used in this experiment. The cameras were installed nearby the rivers by the DID to monitor the real situation on-site. Around 110 images of the DID surveillance cameras were included in the dataset. The complete dataset is composed of 710 images from both the European Flood 2013 Dataset and the DID surveillance cameras. The images included various water-related scenarios to make the training dataset as representative as possible.

All pixels in the images were labeled into two classes, namely water, and background as shown in Figure 1. The labels were used to create ground truth data for training and validation of the semantic segmentation algorithm. The labeling process was carried out by using the Image Labeler tool in the MATLAB® software. The Image Labeler provides a tool to build image databases for computer vision by marking a region of interest from an image collection. The labeling task could be done manually, semi-automatically as well as automatically using an automation algorithm. This study chose to use semi-automatic labeling to ensure the region of interest is defined correctly. As a result, the dataset consists of 710 original images and 710 labeled ground truth. The dataset was divided into a train, validation, and test set. From this dataset, 60% of the images from the dataset were used

for training while the rest of the images were split evenly to 20% and 20% for validation and testing, respectively.

Figure 1. Example of a labeled image. The red region represents the background while the blue region represents waterbodies.

2.2. Semantic Segmentation Architecture

Various deep learning architectures have been developed by previous researchers to solve different tasks such as image classification, object detection, and semantic segmentation. A semantic segmentation network classifies every single pixel in an image that leads to an image that is segmented by class, which can also be seen as a dense-prediction task. To achieve its goal, semantic segmentation generates a segmentation map as the output that has the same size as the input. Semantic segmentation architecture generally consists of the encoder-decoder network. The encoder usually uses pre-trained classification networks that distinguish classes by performing downsampling, while the decoder performs the upsampling operation for learning dense prediction. To illustrate the training procedure, this paper used two different types of convolutional neural network (CNN) that were designed for semantic segmentation tasks, namely DeepLabv3+ [59] and SegNet [60].

2.2.1. DeeplabV3+

The DeepLabv3+ network was invented by the Google Research Team [61], an extended study from DeepLabv1 [62], DeepLabv2 [52], and DeepLabv3 [59]. It is one of the recent state-of-the-art models with many improvements compared to the previous version of DeepLab models. In DeepLabv3+, an effective decoder module was added to recover detailed object boundaries of the segmentation results [61]. In this work, the DeepLabv3+ network was created with weights initialized from a pre-trained ResNet-18 network as a network backbone. ResNet-18 is an efficient network that is well suited for applications with limited processing resources [63].

DeepLabv3+ applies atrous convolutions for upsampling the output of the last convolution layer. The atrous convolution offers a simple yet effective alternative to expand

the field of view of filters without increasing the number of parameters, by adding a rate parameter that corresponds to the stride. It can be expressed as:

$$y[i] = \sum_k x[i + r \cdot k] \cdot w[k] \quad (1)$$

where $y[i]$ is the output, $w[k]$ represents the filter of length k , x is the input feature map, and r represents the atrous rate which corresponds to the stride. Another component in DeepLabv3+ is atrous spatial pyramid pooling (ASPP). It applies atrous convolution in parallel as a method of extracting multi-scale information and improving the accuracy [59]. The encoder features were upsampled by a factor of 4 instead of 16 as in DeepLabv3 [59] and then concatenated with the corresponding low-level features from the encoder network that were convolved by 1×1 convolution. After the concatenation, 3×3 convolutions were applied followed by upsampling by a factor of 4. Figure 2 illustrates the DeepLabv3+ architecture.

Figure 2. An illustration of DeepLabv3+ architecture [61].

2.2.2. SegNet

SegNet consists of encoder-decoder networks with an additional softmax layer for pixel-wise classification as shown in Figure 3. The encoder is composed of 13 convolutional layers which correspond to the first 13 convolution layers in the VGG-16 network [58] that were initially designed for object classification. Nevertheless, the fully connected layers of VGG-16 were discarded to retain higher resolution feature maps and also reduce the number of parameters [60]. The SegNet architecture consists of an encoder followed by a corresponding decoder network which then feeds into a softmax classification layer. Each encoder was connected with the decoder layer. Hence, there were also 13 convolutional layers in the decoder network. In the encoder network, each encoder performed convolutions, where batch normalized and then rectified-linear non-linearity (ReLU) were applied. Next, a 2×2 max-pooling layer with stride 2 was performed that produced the output to be sub-sample by a factor of 2. SegNet architecture differs from other architectures in the manner in which the decoder upsamples its lower resolution input feature maps. The decoder network upsampled its input using the memorized max-pooling indices from the corresponding encoder network to perform non-linear upsampling. Then, the resultant maps were convolved with a trainable decoder filter bank to restore the maps to the original resolution. Finally, a softmax classifier was applied in the final decoder output to predict the class for each pixel. Weights from the pre-trained VGG-16 network were used to initialize the training process in this paper.

Figure 3. An illustration of SegNet architecture [60].

2.2.3. Hyperparameters Configuration

Hyperparameters such as learning rate, epochs, and batch size should be defined before the training procedure. It is crucial to find the best sets of hyperparameters for a specific dataset as there is no correct answer to what set of hyperparameters works for all datasets. The performance of the neural network is highly dependent on the proper settings of a set of parameters [64]. Learning rate, for example, controls the step size for a model to reach the minimum loss function. A model learns faster with a higher learning rate, but it may miss the minimum loss function, which eventually influences the model result. Besides, the number of epochs defines the number of times the entire training dataset is passed through the neural network during training. Therefore, it is important to tune the hyperparameters to optimal values according to the particular dataset. The tuning of the hyperparameters for each of the networks used in this study was discussed in detail in Section 2.3.1.

2.2.4. Data Augmentation

To tackle the problem of having limited training images, data augmentation techniques are frequently used in training deep learning networks [65]. In this work, random reflection and translation techniques were applied to increase the training dataset. The augmentation procedure can provide variation in the training data but at the same time keep the same size of the training sample, which can save the total training time. The augmentation techniques help in avoiding overfitting of the network, which results in better segmentation accuracies even with a small number of the training sample.

2.3. Segmentation Workflow: DeeplabV3+ Versus SegNet

2.3.1. Training Phase

The input images used color images with the resolution of 533 pixels $\times$ 800 pixels for both DeepLabv3+ and SegNet networks. Due to the different network architectures used, the setup parameters of each architecture were fine-tuned differently. For instance, the batch size of each model was adjusted individually considering the GPU limitations. However, the number of epochs in this study was set to be consistent to allow fair comparison. In order to avoid over-fitting, the training task would stop as soon as the loss computed on the validation set degraded over four consecutive epochs.

DeepLabv3+ was trained by applying a stochastic gradient descent with momentum (SGDM) of 0.9 and a value of 0.005 for L2 regularization. The learning rate followed a piecewise schedule that reduced the learning rate from an initial value of 0.0003 by a factor of 0.3 every 10 epochs. A mini-batch size of eight training samples was used and the

model was trained for up to 30 epochs. For SegNet, the network was trained using the same value of SGDM but with a value of 0.0005 for L2 regularization. A mini-batch size of 4 training samples was used instead of 8 as in the DeepLabv3+ to reduce memory usage while training. The network updated the initial learning rate of 0.001 every 10 epochs by multiplying with 0.3 and training was accomplished over 30 epochs. All the models were trained and validated using MATLAB 9.8.0 (R2020a) on a system with the following configurations: Intel® Core™ i7 CPU @ 2.60GHz, 16 GB RAM with NVIDIA GeForce GTX 980M (5GB RAM). Table 1 summarizes the hyperparameters used for each network architecture in this study.

Table 1. Hyperparameters set for this work.

Parameter	DeepLabv3+	Value	SegNet
Momentum		0.9	
Epochs		30	
L2 regularization	0.005		0.0005
Learning rate	0.003		0.001

2.3.2. Inference Phase

The inference phase was carried out to determine how well the trained networks performed when new images were being tested. In order to do that, each of the models was applied to several images that have not been used during the training phase to determine how general the model was.

2.4. Segmentation Evaluation Metrics

The accuracy of the trained models was assessed against the 20% test labeled data as explained in Section 2.1. The results of each algorithm on the test were evaluated using several evaluation metrics including overall accuracy, intersection over union (*IoU*), and boundary F1 score (*BF score*). The metrics were assessed over the entire dataset and per-class version. They were computed using four measures: true positives (*TP*), false positives (*FP*), true negatives (*TN*), and false negatives (*FN*). *TP* was image pixels that were correctly classified as belonging to the water region, while *TN* was the number of pixels that were correctly classified to the non-water (background) class. *FP* was the number of pixels that did not belong to the water region but was wrongly classified as water and *FN* was the pixels that were supposed to be in the water class but were incorrectly associated with the background region.

Overall accuracy is the ratio between the number of pixels that were correctly identified to the total number of pixels without concerning which class the pixels belonged to. It was computed using Equation (2).

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

Intersection over union (*IoU*) or also known as the Jaccard coefficient was the standard measure in evaluating the segmentation results [66]. *IoU* was computed by measuring the overlapping area between the predicted segmentation and the ground truth region. *IoU* is defined by Equation (3).

$$IoU = \frac{TP}{(TP + FP + FN)} \quad (3)$$

Boundary F1 score (*BF score*) was also used to get detailed information on the accuracy of the segmented boundaries as the two above-mentioned metrics provided more region-based accuracies [67]. *BF score* measured how close the boundary of the predicted segmentation and the ground truth segmentation was. It was computed from the precision and recall values considering a distance error tolerance over the boundary pixels [68].

Precision is the number of correctly classified positive results divided by all positive results and recall is the number correctly classified positive results divided by the number of samples that should have been classified as positive. The mathematical representation of the *BF score* is defined as in Equation (4).

$$BF\ score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (4)$$

2.5. Water Level Observation

2.5.1. Study Area

A surveillance camera was installed at Paya Besar, Kuantan, Pahang with latitude and longitude coordinates of 3°47'28" N and 103°14'49" E, respectively (Figure 4). Images from five different days, from 3 January to 7 January 2021, were used to detect water body regions using one of the two semantic segmentation networks that demonstrate better performance than the other during the training. One image for each day was used so that the significant difference in river water levels could be shown from the results. In addition, the weather conditions during these periods were wet, where some days were raining heavily and some were not. Thus, the rising and falling of water levels can be demonstrated more clearly. Nonetheless, there is no water level sensor available in this area. Therefore, before applying the water level framework in this study, the framework was tested in another area with a water level sensor to validate the performance of the proposed work.

Figure 4. The yellow camera icon represents the location of the surveillance camera that was used for water detection using semantic segmentation (Image courtesy: Google Earth).

2.5.2. Water Level Framework

A deep learning architecture network that obtained higher accuracy results was then be used for flood detection applications. The model was applied to images captured by a surveillance camera that was installed in the study area. Water body regions from the segmentation results were extracted and converted into binary masks before undergoing the post-processing stage to enhance the meaningful feature. The first morphological operation was to remove small objects from the binary image. It is assumed that the water region was one connected component, hence, small objects with less than 100,000 pixels were erased. Next, morphological closing was performed on the binary image to smooth the edges and close any open holes that were specified at a radius of 2 pixels. The last morphological operation is to fill holes in the interior of the water feature.

In order to obtain water level information, terrestrial light detection and ranging (LiDAR) was used in this study as it can provide a high-resolution digital elevation model (DEM) and has been widely applied in flood studies [26]. Field data collection for LiDAR was done once during the normal water condition (Figure 5). Several elevation points were extracted from LiDAR data as water level markers based on on-site references during data collection and the marker locations were also identified in surveillance images prior to the segmentation process. The extraction of elevation values was done in geographical information system (GIS) environment by using ArcGIS 10.2.2 software. To replicate this work, it is suggested to replace LiDAR data with a global navigation satellite system (GNSS) with centimeter accuracy. The elevation values of the markers were then overlaid with the post-processed water region to get the idea of how depth the water levels are as shown in Figure 5. Four water level markers represent different water level thresholds used in this study. Green 'x' represents normal level, yellow 'x' represents alert level, and orange and red 'x' represent warning and danger levels, respectively, and elevation for each point was identified.

Figure 5. Water level markers were extracted from LiDAR data during normal conditions.

River water levels were estimated based on the elevation from each marker by computing the average elevations of two water thresholds, if possible. For instance, if the water region covered the area between normal and alert markers, the mean elevations of both water thresholds were calculated to estimate the water level for that particular condition. If the water region was covered up to only one marker, then the water level followed the elevation value of the particular marker. On the other hand, when the water region was less than the normal threshold or exceeded the danger threshold, this study did not have absolute values yet, as there was limited information regarding the elevations at these levels during the validation stage.

Since there is no water level sensor in the study area, the proposed approach was used in another location equipped with a water level sensor to validate the water level estimation. The relationship between the estimated water level and water level measured by the sensor was evaluated by using Spearman's rank-order correlation coefficient [69]. It is a nonparametric test to identify the strength and direction of a monotonic relationship between two sets of continuous or ordinal data. Equation (5) was used for computing the rank correlation with ties rank.

$$\rho = 1 - 6 \left[\frac{\sum D^2 + \frac{m_i^3 - m_i}{12} + \dots}{n(n^2 - 1)} \right] \quad (5)$$

where D is the difference between ranks, n is the number of pairs of data, and m_i is the number of repetitions of i th rank. Spearman's correlation coefficient ranges from +1 to −1, with a correlation closer to 0 indicating a weaker association between the two variables.

3. Results

3.1. Water Segmentation

3.1.1. Performance Evaluation

Semantic segmentation applying DeepLabv3+ and SegNet approaches was performed to classify images into the water body and background regions. Both segmentation architectures were trained using the training dataset as described in Section 2.1. The training time varied depending on the complexity of the network and the number of features extracted. To evaluate the performances of the trained networks, testing was accomplished using 20% of the original dataset, taking a total of 142 images. In particular, SegNet was more complex as compared to DeepLabv3+, which over-limit the GPU capacity causing the network to be trained on CPU. Hence, the training time for SegNet was significantly longer than DeepLabv3+. Therefore, this study did not compare the performance of these two networks in terms of time-wise. Rather, they were evaluated in terms of overall accuracy, IoU, and BF score for the entire dataset and individual classes as well as visual comparison of the segmentation results. The accuracy results of the predicted water body extracted from the segmentation task for the aggregate dataset are shown in Figure 6 and the performance of per-class metrics is illustrated in Figures 7 and 8.

Figure 6 summarizes the overall dataset accuracy for the semantic segmentation achieved by the two trained models. Both models achieved an overall accuracy of more than 95% with DeepLabv3+ obtaining the highest value of 96.72%. For the IoU metric, DeepLabv3+ achieved 94.12% while SegNet achieved 91.05%, indicating that there is a high agreement between the predicted and referenced labeled data. However, the segmentation performance in terms of BF score is not as high as overall accuracy and IoU metrics. DeepLabv3+ obtained 82.40%, which is better than the result obtained by SegNet that achieved only 67.20%. The results demonstrate that the matching boundary between the predicted and ground truth region was not as satisfactory as expected.

Figure 6. A comparison of segmentation metrics for the entire dataset for both DeepLabv3+ and SegNet.

Figure 7. Class-specific metrics for DeepLabv3+ network.

Figure 8. Class-specific metrics for SegNet network.

Figure 7 shows that the class-specific accuracy for the segmentation of water bodies from CCTV images obtained by the trained model when using the DeepLabv3+ network. It demonstrates that the performance metrics of DeepLabv3+ are slightly better than SegNet (Figure 8). The overall accuracy for both water and background classes was achieved above 95% for DeepLabv3+. In terms of the IoU metric, the DeepLabv3+ trained model achieved around 92.08% and 95.31% for water and background classes, respectively. DeepLabv3+ outperformed SegNet for the BF score metric by achieving 79.22% for water and 85.59% for background class.

For per class performance, SegNet achieved approximately 94% to 96% of overall accuracy for both classes with the IoU metric obtained around 88.82% and 93.29% for water and background classes, respectively as shown in Figure 8. The result of the BF score was low as compared to other metrics with the water class only obtaining around 66.44% and the background class achieving 67.96%.

Although the overall dataset performance is high, the BF score for the class metric shows that the boundaries of water were poorly mapped to the correct contours, especially for SegNet. This might happen due to the class imbalance problem, where the number of pixels that belonged to the water class was smaller than the background class [45].

Additional data that includes more pixels in the water class might help to improve the segmentation results. Based on the results obtained, it clearly shows that the model trained using the DeepLabv3+ network outperformed the one that used SegNet in terms of accuracy, IoU, and BF score metrics.

3.1.2. Visual Analysis of the Segmentation Results

To visualize the segmentation results more clearly, this work represents the outcomes of the testing images. The original and the ground truth images are also presented to compare with the segmentation outcomes. Figure 9 shows the outcomes of DeepLabv3+ and SegNet models for the two sample images of the test dataset. Both networks could identify water regions in most of the images, even though the color of water regions vary from one another. However, it can be seen from Figure 9c,d that some of the water pixels were misidentified as background pixels due to similar color features between the water region and objects. Overall, the segmentation results from the DeepLabv3+ network yielded more accurate predictions as it could segment small objects such as signboards from water regions while SegNet failed to do so.

Figure 9. Visualization and comparison of two representative results on test data. (a) Original image. (b) Ground truth. (c) DeepLabv3+ prediction. (d) SegNet prediction. Pink and green colored regions represent water and background, respectively.

To see the errors more distinctly, the segmentation results were compared with the expected ground truth. The magenta and green regions highlight FN and FP areas, the areas of segmented results that differed from the expected ground truth as shown in Figure 10. The visual inspection shows that semantic segmentation results overlapped quite well for water class especially for DeepLabv3+, but the boundary of the segmented regions is not as accurate, which led to low BF scores. Besides, the figure also demonstrates that segmentation results using DeepLabv3+ obtained fewer errors as compared to SegNet.

Figure 10. Comparisons between segmentation results and ground truth data are illustrated by overlay columns. (a) The original images (b) DeepLabv3+ prediction and ground truth (c) SegNet prediction and ground truth.

3.2. Inference Phase

During the inference phase, the trained models were tested on several images captured from four surveillance cameras located at different spots in Selangor. In general, both trained networks could perform semantic segmentation tasks when introduced to new images but DeepLabv3+ could separate water elements from the background better than SegNet. DeepLabv3+ obtained more accurate water segmentation and smooth water regions delineation as compared to SegNet as shown in Figure 11.

Figure 11. Visualization and comparison of segmentation results during inference phase. (a) Original image. (b) DeepLabv3+ prediction. (c) SegNet prediction.

3.3. Comparison with Previous Findings

Results from this work were comparable with a highly similar work done by Lopez-Fuentes et al., (2017) that used three different segmentation algorithms, namely fully convolutional network (FCN-8s), fully convolutional DenseNet (Tiramisu), and Pix2Pix networks, as well as Vandaele et al., (2021) who applied Deeplab and UperNet networks for semantic segmentation. The trained models were then applied to several images with different backgrounds instead of just one specific location. Mean IoU was used to evaluate the segmentation results since both authors included IoU evaluation in their works. For the study carried out by Vandaele et al., (2021), only the highest IoU results were included. The comparison between findings from this work and the previous study is reported in Table 2. The comparison shows that the proposed approach (DeepLabV3+) obtained higher IoU results over the work of Lopez-Fuentes et al. [29] and Vandaele et al. [70].

Table 2. A comparison between results from different segmentation networks used in this work and the previous study.

Author(s)	Segmentation Networks	IoU (%)
Proposed approach	DeepLabv3+	94.37
	SegNet	91.05
Vandaele et al. [70]	DeepLab	93.74
	UperNet	93.32
Lopez-Fuentes et al. [29]	Tiramisu	81.91
	Pix2pix	72.25
	FCN-8s	70.05

3.4. Practical Application for Water Level Observation

Before applying the river water level estimation in the study area, the work concept was used in one of the DID stations, Sungai Bernam at Kampung Selisek Selangor, that equipped with both a surveillance camera and water level sensor to validate its efficiency. For this particular location, the water level thresholds for normal, alert, warning, and danger are 24 m, 26.1 m, 26.6 m, and 27.1 m, respectively. Statistically, the relationship between both estimated water level and observed water level from the sensor was evaluated using Spearman's rank-order correlation coefficient. It is found that the correlation between the two variables was 0.91, which indicates that the correlation is strong. Therefore, the proposed method could be used for application in the study area with no water level sensor available.

Visually, Figure 12 shows the comparison between water level estimation from the proposed work and water level obtained from the sensor. In general, the estimated water level follows the fluctuation pattern measured by the sensor. However, it can be seen that there were a few points of estimated water level that were significantly different from the one measured by the sensor, especially within the normal range. This is because the range gap between normal and alert levels is quite far, and the study has markers only at normal and alert points with no other points in between. Hence, the large errors between both measurements. Therefore, more elevation points are needed to reduce the errors between the estimated and observed water levels. Nevertheless, it is more important to focus on alert, warning, and danger levels. For the three categories, the proposed method tends to overestimate the water level at some points, but still within the proposed categories. Nonetheless, more obvious errors were found when the method underestimated the water level.

It is found that errors in water segmentation results were the major factor that contributes to this problem. If the segmentation model failed to identify the water region correctly, it led to errors during water level estimation. For instance, the water region in Figure 13 was not segmented correctly, which results in incorrect water level estimation. The water level in Figure 13 should be at the danger level, but it was identified as a warning level. This problem can be avoided by preparing more training datasets for the segmentation model to improve the segmentation accuracy.

Figure 12. Comparison between the estimated water level and observed water level measured by the sensor. The blue line indicates water level from the sensor while the black line represents water level estimation from the proposed method. Water level thresholds for normal level, alert level, warning level, and danger level were represented by green, yellow, orange, and red colors, respectively.

Figure 13. Failure in water detection during the segmentation process led to an error in water level estimation. (a) The segmentation result overlaid with the original image. (b) The estimated river water level and its threshold class.

Based on the findings, the DeepLabv3+ model was used to segment a set of surveillance images in the study area captured at five consecutive days. Water regions were detected before the integration with water level information from LiDAR data to see the fluctuation of water level as well as to estimate water level in the study area. After the segmentation process took place, a binary mask for the water region was created and the morphological operations were carried out to remove noises and imperfections in the segmentation outputs, as shown in Figure 14.

Figure 14. (a) The original images were captured on different days (b) The segmentation results obtained when using DeepLabv3+ model (c) Results after performing morphological operations.

From the results obtained, it is clearly observed that the model overestimated water features at the lower right side of the images (Figure 14b), especially before the application of morphological operations. This is due to the fact that there was a brownish ground area that had a similar attribute as the water feature that led to the misclassifications. Besides, the image captured on 6 January obtained more error as compared to the other images because there were water stains on the camera lens during rain that was mistakenly classified as water region. Thereby, the two factors have a negative influence on the segmentation results.

To improve these results, morphological operations were applied and some unwanted regions were removed successfully as shown in Figure 14c. However, there were some parts of water that were also removed during these morphological operations. This is because, as the water region was assumed to be one connected element in the image, water features that were disconnected from the main water element were presumed to be noises. This problem might be avoided if there were no obstacles that separate all water

regions present in one particular image or different morphological operations, such as morphological dilation and erosion, were adopted during the post-processing stage.

On a side note, elevation values of the on-site markers were extracted from the digital elevation model (DEM) generated from LiDAR data. A geoprocessing tool in ArcGIS software was executed to extract the cell values of the DEM raster based on a set of marker points. The elevation values of the markers were then overlaid with the post-processed water region to get an idea of how water levels fluctuated during the wet condition. There were four water level thresholds used in this study, which are normal, alert, warning, and danger levels. In particular, a green color marker represents normal level, a yellow color marker indicates alert level, while orange and red markers represent warning and danger levels, respectively, as shown in Figure 15.

Figure 15. The segmentation results were overlaid with the water level thresholds extracted from LiDAR data to give clear visualization on the rising and falling of water levels during the wet condition. The green marker represents the normal level, the yellow marker represents the alert level, the orange level indicates the warning level and the red marker indicates the danger level.

River water level estimation was carried out using the framework discussed in Section 2.5.2. After evaluating the performance of the proposed framework, it was applied to the study area which does not equip with a water level sensor. The threshold values for this study area were set at 8.8 m for normal level, 9.3 m for alert level, 9.8 m for warning level, and 10.8 m for dangerous level, respectively (Figure 15). Specifically, on 3 January, the water level exceeded the danger threshold which indicates that the water level value was 10.8 m. Moreover, the danger level demonstrates that possible flooding

may occur and evacuation action may be initiated. On the next day, the water level dropped to alert level at 9.55 m and fell again to normal level on the 5 of January, which means the water level value was estimated to be 9.05 m. Due to rainfall on 6 and 7 January, the water level rose to warning level which demonstrates that water levels were significantly increased above the normal value during these days. The water level value for 6 January was 10.3 m while on 7 January it was estimated at 9.55 m.

4. Discussion

Flood monitoring and management mostly rely on data obtained from in-situ sensor measurements such as rainfall and water level sensors. The means provide almost real-time transmission of one-dimensional measurement data at a high level of accuracy. However, one spatial dimension values are not enough to represent the spatial dynamics of the surface water extent. The lack of dynamic information leads to the difficulty in understanding the field situation by decision-makers who need to make proper plans and take appropriate actions in reducing the impacts of flood events. This work proposed to apply semantic segmentation methods for water body detection using images captured by surveillance cameras. By using these methods, a static surveillance camera could be changed into an active visual sensor that brings benefits, especially in analyzing the real situation on-site in the context of flood disasters.

Semantic segmentation architecture is generally composed of encoder and decoder networks, where encoder performs downsampling to distinguish classes while decoder performs the upsampling operation to learn dense prediction. In particular, two state-of-the-art segmentation networks were employed in this work, which are SegNet and a recent segmentation network known as DeepLabv3+ networks. These two networks were selected based on their superior performances demonstrated in previous studies [71–76]. In a recent study, Khan et al. examined several semantic segmentation models including SegNet and DeepLabv3+ for smoke segmentation. It was reported that both models achieved competitively promising performances and obtained better evaluation results as compared to other segmentation models. Hence, this work exclusively discussed the two semantic segmentation models. The findings show that both trained models could be used in water segmentation but the model trained using the DeepLabv3+ network outperformed the one that used SegNet in terms of evaluation metrics. In particular, DeepLabv3+ obtained more accurate water segmentation and smooth water regions delineation as compared to SegNet. The results aligned with previous studies as DeepLabv3+ consists of an effective decoder module that helps in retrieving detailed object boundaries of the segmentation results [61,77–79]. However, when applying the DeepLabv3+ model to a series of five-day images in the study area, the results were not as neat as segmentation results in other locations. The presence of similar properties as water regions and background aspects might influence the segmentation results, which were then improved by performing post-processing steps. Besides, water stains on the lens during rainfall can also contribute to poor segmentation outcomes [80].

To visually examine the rising and falling of water levels, water level values were extracted from LiDAR data based on on-site references and the reference locations were identified in the surveillance image as well. This work proposed to use water level markers from LiDAR data as a threshold value reference to avoid error when estimating water levels using target objects that are present in the study area [81]. On-site data collection was undertaken during normal water conditions so that several water level thresholds could be adopted including the normal water level. Four water level thresholds namely normal level, alert level, warning level, and danger level were employed, to follow the standard water level thresholds used by DID. The results obtained illustrate the fluctuation of water levels in the study area during raining season.

The relationship between water level estimation using the proposed framework and water level measurement from the sensor was investigated using Spearman's rank-order correlation coefficient. Both variables proved to have a strong correlation. Therefore, the

concept was applied to the study area that has no water level sensor available. Besides estimating the water level, the proposed method can also provide information in terms of water level thresholds (normal level, alert level, warning level, or danger level) simultaneously, which can act as an added value that aids the authorities to quickly understand the current situation and detect any changes in water level visually to avoid uncertainty in decision making. There were several problems detected when analyzing the proposed method, such as some segmentation errors which may be the result of the lack of a training dataset or poor choice of hyperparameters during the training stage. In terms of water level estimation, providing more elevation points within the water level thresholds may reduce the estimated errors and increase the efficiency of the proposed framework. The current results show that there is plenty of room for improvement regarding the proposed method.

Compared to the current approach of measuring water levels in rivers using water level sensors such as pressure transducer, radar, ultrasonic, and air bubbler, the proposed water level framework has a high potential to provide water level information at a low cost. In general, the installation of water level sensors can cost around RM13,000 (roughly around \$3100), not including the maintenance cost. The pressure sensor needs to be submerged in the water, which results in high maintenance costs due to river sedimentation or damage by flood. Installation of radar and ultrasonic is limited as they need to be set up on structures, such as bridges, footbridges, and jibs. The air bubble is suitable to be installed in the river with deep slope elevation and the sensor is not in direct contact with the water. The only part of the sensor that is in contact with the water is the bubble tube but it often requires frequent maintenance. In contrast, this work proposed to use a surveillance camera, which is much cheaper in terms of cost and easy to maintain. The installation cost of one surveillance camera is between RM1500 and RM2000 (around RM\$360–\$600), depending on the features offered, with straightforward installation. Using the surveillance images, information regarding the flood extent was extracted using deep learning techniques, which DeepLabv3+ demonstrated better performance compared to SegNet, and then the segmentation results were further analyzed using water level framework (Section 2.5.2) to estimate water levels and their category. Besides, the proposed approach can observe the water fluctuation more clearly, which can be easily understood by the authorized personnel, and help in response planning.

5. Conclusions

This work presented state-of-the-art semantic segmentation methods using two different networks, namely DeepLabv3+ and SegNet, for detecting water bodies before estimating water levels and observing water fluctuation from surveillance images. DeepLabv3+ was created with ResNet as the backbone, while SegNet used VGG-16, and a comparison between these two approaches has been done in this work. Both networks could distinguish features of the water from the background with some misidentification of pixels due to similar features. This work focused on the accuracy factor only without considering the training time and memory usage. When measuring semantic segmentation metrics on the overall test set results, it was found that the DeepLabv3+ network performed better than the SegNet network by achieving above 93% for overall accuracy and IoU metrics, and approximately 82% for BF score while SegNet achieving averages of 91% and 67.20% respectively. For individual class performance, DeepLabv3+ still outperformed SegNet by having higher overall accuracy, IoU, and BF score. When predicting new images using both trained networks, the results show that both networks successfully distinguished water regions from the background. However, the outputs from DeepLabv3+ were more accurate than the results from the SegNet network. Therefore, DeepLabv3+ was used in the segmentation process for a set of images collected for five straight days to estimate river water levels and determine the rising and falling (fluctuation) of water levels in the study area.

When both segmentation results and water level markers were overlaid, the fluctuation of water levels can be clearly visualized and water level threshold can be obtained, hence it is assumed that it could help the authorities to make a clear judgment in planning and decision making. Water level estimation was predicted based on the elevations from the pre-defined markers. The values were estimated either from the exact elevation value of the marker or by computing the average values between the two markers. To investigate the performance of the proposed method, the relationship between the estimated water level and water level measured by the sensor was evaluated by using Spearman's rank-order correlation coefficient. The correlation obtained was 0.91 which indicates a strong relationship between the water level measured by the sensor and water level estimated from the proposed framework.

Based on these findings, it has been concluded that the proposed approach is a promising technique that can be employed to extract water regions as well as estimate water levels from surveillance images for flood studies. By identifying the flooded regions, it is believed that the information might help the authorities to focus on the most vulnerable regions during flood events. Moreover, by combining the segmentation results with other data such as LiDAR data, it is expected that the outputs could provide additional information for flood management and related activities. The proposed study has the potential as an alternative monitoring system that offers information such as precise locations and conditions of the affected area at a low cost. Furthermore, if the telemetry system failed to provide data due to severe flooding, it can act as a substitute solution because the proposed work could also provide the level of severity of the flood as well as approximate water level.

Future work could be done by having an extensive labeled dataset and hyperparameter fine-tuning to improve the segmentation accuracy. In addition, the same work could also be explored with a nighttime dataset to determine how well the neural network functions when using images that are captured at night. Furthermore, an extensive study on extracting more information from flooding images, such as calculating the flood depth and flooded area, is ongoing, which it is believed can be done in a GIS environment.

Author Contributions: N.A.M.: conceptualization, formal analysis, methodology, writing—original draft, visualization. A.F.A.: funding acquisition, supervision, visualization, writing—review, editing. S.K.B.: software, supervision. M.R.M.: supervision. A.M.: supervision. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by Universiti Putra Malaysia under the Putra Grant, GPB (Grant number: 9678700).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available data were analyzed in this study. This data can be found here: <http://www.inf-cv.uni-jena.de/Research/Datasets/European+Flood+2013.html> (accessed on 20 July 2021) and <http://infobanjirjps.selangor.gov.my/camera.html> (accessed on 15 December 2020).

Acknowledgments: The authors appreciate the support for this study from the Universiti Putra Malaysia and the Institute of Aquaculture and Aquatic Sciences. The authors would like to thank the Department of Irrigation and Drainage for sharing the surveillance images and water level data publicly.

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

References

1. Demski, C.; Capstick, S.; Pidgeon, N.; Sposato, R.G.; Spence, A. Experience of extreme weather affects climate change mitigation and adaptation responses. *Clim. Chang.* **2017**, *140*, 149–164. [CrossRef]
2. Wan Ruslan Ismail, T.H. Extreme weather and floods in Kelantan state, Malaysia in December 2014. *Res. Mar. Sci.* **2018**, *3*, 231–244.

3. Azimi, M.A.; Syed Zakaria, S.A.; Majid, T.A. Disaster risks from economic perspective: Malaysian scenario. *IOP Conf. Ser. Earth Environ. Sci.* **2019**, *244*, 012009. [\[CrossRef\]](#)
4. Blunden, J.; Boyer, T. State of the Climate in 2020. *Bull. Amer. Meteor. Soc.* **2021**, *102*, 1–475. [\[CrossRef\]](#)
5. UNDRR. *The Human Cost of Disasters—An Overview of the Last 20 Years 2000–2019*; UNDRR: Geneva, Switzerland, 2020.
6. Matgen, P.; Schumann, G.; Henry, J.B.; Hoffmann, L.; Pfister, L. Integration of SAR-derived river inundation areas, high-precision topographic data and a river flow model toward near real-time flood management. *Int. J. Appl. Earth Obs. Geoinf.* **2007**, *9*, 247–263. [\[CrossRef\]](#)
7. Mason, D.C.; Davenport, I.J.; Neal, J.C.; Schumann, G.J.P.; Bates, P.D. Near real-time flood detection in urban and rural areas using high-resolution synthetic aperture radar images. *IEEE Trans. Geosci. Remote Sens.* **2012**, *50*, 3041–3052. [\[CrossRef\]](#)
8. Skakun, S.; Kussul, N.; Shelestov, A.; Kussul, O. Flood Hazard and Flood Risk Assessment Using a Time Series of Satellite Images: A Case Study in Namibia. *Risk Anal.* **2014**, *34*, 1521–1537. [\[CrossRef\]](#) [\[PubMed\]](#)
9. García-pintado, J.; Mason, D.C.; Dance, S.L.; Cloke, H.L.; Neal, J.C.; Freer, J.; Bates, P.D. Satellite-supported flood forecasting in river networks: A real case study. *J. Hydrol.* **2015**, *523*, 706–724. [\[CrossRef\]](#)
10. Huang, X.; Wang, C.; Li, Z. GIS A near real-time flood-mapping approach by integrating social media and post-event satellite imagery. *Ann. GIS* **2018**, *24*, 113–123. [\[CrossRef\]](#)
11. DID. *DID Manual (Volume 1—Flood Management)*; Jabatan Pengairan dan Saliran Malaysia: Kuala Lumpur, Malaysia, 2009; Volume 1.
12. Marin-Perez, R.; García-Pintado, J.; Gómez, A.S. A real-time measurement system for long-life flood monitoring and warning applications. *Sensors* **2012**, *12*, 4213–4236. [\[CrossRef\]](#) [\[PubMed\]](#)
13. Ji, Y.; Zhang, M.; Wang, Y.; Wang, P.; Wang, A.; Wu, Y.; Xu, H.; Zhang, Y. Microwave-Photonic Sensor for Remote Water-Level Monitoring Based on Chaotic Laser. *Int. J. Bifurc. Chaos* **2014**, *24*, 1450032. [\[CrossRef\]](#)
14. Lo, S.W.; Wu, J.H.; Lin, F.P.; Hsu, C.H. Cyber surveillance for flood disasters. *Sensors* **2015**, *15*, 2369–2387. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Kang, W.; Xiang, Y.; Wang, F.; Wan, L.; You, H. Flood detection in gaofen-3 SAR images via fully convolutional networks. *Sensors* **2018**, *18*, 2915. [\[CrossRef\]](#)
16. Yang, C.; Wei, Y.; Wang, S.; Zhang, Z.; Huang, S. Extracting the flood extent from satellite SAR image with the support of topographic data. In *Proceedings of the 2001 International Conferences on Info-Tech and Info-Net: A Key to Better Life, ICII 2001—Proceedings, Beijing, China, 29 October–1 November 2001; Volume 1*, pp. 87–92.
17. Cao, H.; Zhang, H.; Wang, C.; Zhang, B. Operational flood detection using Sentinel-1 SAR data over large areas. *Water* **2019**, *11*, 786. [\[CrossRef\]](#)
18. Jung, H.C.; Jasinski, M.; Kim, J.W.; Shum, C.K.; Bates, P.; Neal, J.; Lee, H.; Alsdorf, D. Calibration of two-dimensional floodplain modeling in the central Atchafalaya Basin Floodway System using SAR interferometry. *Water Resour. Res.* **2012**, *48*, 1–13. [\[CrossRef\]](#)
19. Schumann, G.J.P.; Neal, J.C.; Mason, D.C.; Bates, P.D. The accuracy of sequential aerial photography and SAR data for observing urban flood dynamics, a case study of the UK summer 2007 floods. *Remote Sens. Environ.* **2011**, *115*, 2536–2546. [\[CrossRef\]](#)
20. Feng, Q.; Liu, J.; Gong, J. Urban flood mapping based on unmanned aerial vehicle remote sensing and random forest classifier-A case of yuyao, China. *Water* **2015**, *7*, 1437–1455. [\[CrossRef\]](#)
21. Perks, M.T.; Russell, A.J.; Large, A.R.G. Technical note: Advances in flash flood monitoring using unmanned aerial vehicles (UAVs). *Hydrol. Earth Syst. Sci.* **2016**, *20*, 4005–4015. [\[CrossRef\]](#)
22. Langhammer, J.; Bernsteinová, J.; Miřijovský, J. Building a high-precision 2D hydrodynamic flood model using UAV photogrammetry and sensor network monitoring. *Water* **2017**, *9*, 861. [\[CrossRef\]](#)
23. Lindner, G.; Schraml, K.; Mansberger, R.; Hübl, J. UAV monitoring and documentation of a large landslide. *Appl. Geomatics* **2016**, *8*, 1–11. [\[CrossRef\]](#)
24. Nishar, A.; Richards, S.; Breen, D.; Robertson, J.; Breen, B. Thermal infrared imaging of geothermal environments and by an unmanned aerial vehicle (UAV): A case study of the Wairakei—Tauhara geothermal field, Taupo, New Zealand. *Renew. Energy* **2016**, *86*, 1256–1264. [\[CrossRef\]](#)
25. Wierzbicki, D.; Kedzierski, M.; Fryskowska, A. Assessment of the influence of UAV image quality on the orthophoto production. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2015**, *40*, 1–8. [\[CrossRef\]](#)
26. Muhadi, N.A.; Abdullah, A.F.; Bejo, S.K.; Mahadi, M.R.; Mijic, A. The Use of LiDAR-Derived DEM in Flood Applications: A Review. *Remote Sens.* **2020**, *12*, 2308. [\[CrossRef\]](#)
27. Filonenko, A.; Hernandez, D.C.; Seo, D.; Jo, K.H. (Eds.) Real-time flood detection for video surveillance. In *Proceedings of the IECON 2015—41st Annual Conference of the IEEE Industrial Electronics Society, Yokohama, Japan, 9–12 November 2015*; IEEE: Piscataway Township, NJ, USA, 2015; pp. 4082–4085.
28. Muhadi, N.A.; Abdullah, A.F.; Bejo, S.K.; Mahadi, M.R.; Mijic, A. Image Segmentation Methods for Flood Monitoring System. *Water* **2020**, *12*, 1825. [\[CrossRef\]](#)
29. Lopez-Fuentes, L.; Rossi, C.; Skinnemoen, H. River segmentation for flood monitoring. In *Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017*; IEEE: Piscataway Township, NJ, USA, 2017; pp. 3746–3749. [\[CrossRef\]](#)
30. Tsubaki, R.; Fujita, I.; Tsutsumi, S. Measurement of the flood discharge of a small-sized river using an existing digital video recording system. *J. Hydro-Environ. Res.* **2011**, *5*, 313–321. [\[CrossRef\]](#)

31. Guillén, N.F.; Patalano, A.; García, C.M.; Bertoni, J.C. Use of LSPIV in assessing urban flash flood vulnerability. *Nat. Hazards* **2017**, *87*, 383–394. [\[CrossRef\]](#)
32. Costa, D.G.; Guedes, L.A.; Vasques, F.; Portugal, P. Adaptive monitoring relevance in camera networks for critical surveillance applications. *Int. J. Distrib. Sens. Netw.* **2013**, *9*, 836721. [\[CrossRef\]](#)
33. Arshad, B.; Ogie, R.; Barthelemy, J.; Pradhan, B.; Verstaavel, N.; Perez, P. Computer vision and iot-based sensors in flood monitoring and mapping: A systematic review. *Sensors* **2019**, *19*, 5012. [\[CrossRef\]](#) [\[PubMed\]](#)
34. Nath, R.K.; Deb, S.K. Water-Body Area Extraction From High Resolution Satellite Images—An Introduction, Review, and Comparison. *Int. J. Image Process.* **2010**, *3*, 353–372.
35. Geetha, M.; Manoj, M.; Sarika, A.S.; Mohan, M.; Rao, S.N. Detection and estimation of the extent of flood from crowd sourced images. In *Proceedings of the 2017 International Conference on Communication and Signal Processing (ICCSP), Melmaruvathur, India, 6–8 April 2017*; IEEE: Piscataway Township, NJ, USA, 2017; pp. 603–608.
36. Jyh-Horng, W.; Chien-Hao, T.; Lun-Chi, C.; Shi-Wei, L.; Fang-Pang, L. Automated Image Identification Method for Flood Disaster Monitoring In Riverine Environments: A Case Study in Taiwan. In *Proceedings of the AASRI International Conference on Industrial Electronics and Applications (IEA 2015), London, UK, 27–28 June 2015*; Atlantis Press: London, UK, 2015; pp. 268–271.
37. Lo, S.W.; Wu, J.H.; Lin, F.P.; Hsu, C.H. Visual sensing for urban flood monitoring. *Sensors* **2015**, *15*, 20006–20029. [\[CrossRef\]](#) [\[PubMed\]](#)
38. Buscombe, D.; Ritchie, A.C. Landscape classification with deep neural networks. *Geosciences* **2018**, *8*, 244. [\[CrossRef\]](#)
39. Wang, L.; Zhang, J.; Liu, P.; Choo, K.K.R.; Huang, F. Spectral-spatial multi-feature-based deep learning for hyperspectral remote sensing image classification. *Soft Comput.* **2017**, *21*, 213–221. [\[CrossRef\]](#)
40. Yuan, Q.; Shen, H.; Li, T.; Li, Z.; Li, S.; Jiang, Y.; Xu, H.; Tan, W.; Yang, Q.; Wang, J.; et al. Deep learning in environmental remote sensing: Achievements and challenges. *Remote Sens. Environ.* **2020**, *241*, 111716. [\[CrossRef\]](#)
41. Tapu, R.; Mocanu, B.; Zaharia, T. DEEP-SEE: Joint object detection, tracking and recognition with application to visually impaired navigational assistance. *Sensors* **2017**, *17*, 2473. [\[CrossRef\]](#) [\[PubMed\]](#)
42. Feng, J.; Wei, Y.; Tao, L.; Zhang, C.; Sun, J. Salient object detection by composition. In *Proceedings of the 2011 International Conference on Computer Vision, Barcelona, Spain, 6–13 November 2011*; pp. 1028–1035. [\[CrossRef\]](#)
43. Hawari, A.; Alamin, M.; Alkadour, F.; Elmasry, M.; Zayed, T. Automated defect detection tool for closed circuit television (cctv) inspected sewer pipelines. *Autom. Constr.* **2018**, *89*, 99–109. [\[CrossRef\]](#)
44. Varfolomeev, I.; Yakimchuk, I.; Safonov, I. An application of deep neural networks for segmentation of microtomographic images of rock samples. *Computers* **2019**, *8*, 72. [\[CrossRef\]](#)
45. Steccanella, L.; Bloisi, D.D.; Castellini, A.; Farinelli, A. Waterline and obstacle detection in images from low-cost autonomous boats for environmental monitoring. *Rob. Auton. Syst.* **2020**, *124*, 103346. [\[CrossRef\]](#)
46. Tong, X.Y.; Xia, G.S.; Lu, Q.; Shen, H.; Li, S.; You, S.; Zhang, L. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sens. Environ.* **2020**, *237*, 111322. [\[CrossRef\]](#)
47. Waldner, F.; Diakogiannis, F.I. Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network. *Remote Sens. Environ.* **2020**, *245*, 111741. [\[CrossRef\]](#)
48. Bischke, B.; Bhardwaj, P.; Gautam, A.; Helber, P.; Borth, D.; Dengel, A. Detection of Flooding Events in Social Multimedia and Satellite Imagery using Deep Neural Networks. In *Proceedings of the MediaEval, Dublin, Ireland, 13–15 September 2017*; Volume 1984, pp. 13–15.
49. Yu, L.; Wang, Z.; Tian, S.; Ye, F.; Ding, J.; Kong, J. Convolutional Neural Networks for Water Body Extraction from Landsat Imagery. *Int. J. Comput. Intell. Appl.* **2017**, *16*, 1750001. [\[CrossRef\]](#)
50. Amit, S.N.K.B.; Aoki, Y. Disaster detection from aerial imagery with convolutional neural network. In *Proceedings of the Proceedings—International Electronics Symposium on Knowledge Creation and Intelligent Computing, IES-KCIC 2017, Surabaya, Indonesia, 26–27 September 2017*; pp. 239–245.
51. Isikdogan, F.; Bovik, A.C.; Passalacqua, P. Surface water mapping by deep learning. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2017**, *10*, 4909–4918. [\[CrossRef\]](#)
52. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [\[CrossRef\]](#)
53. Feng, W.; Sui, H.; Huang, W.; Xu, C.; An, K. Water Body Extraction from Very High-Resolution Remote Sensing Imagery Using Deep U-Net and a Superpixel-Based Conditional Random Field Model. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 618–622. [\[CrossRef\]](#)
54. Li, Y.; Martinis, S.; Wieland, M. Urban flood mapping with an active self-learning convolutional neural network based on TerraSAR-X intensity and interferometric coherence. *ISPRS J. Photogramm. Remote Sens.* **2019**, *152*, 178–191. [\[CrossRef\]](#)
55. Moy de Vitry, 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. Discuss.* **2019**, *23*, 4621–4634. [\[CrossRef\]](#)
56. Akiyama, T.S.; Marcato Junior, J.; 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.* **2020**, *43*, 1189–1193. [\[CrossRef\]](#)
57. Vandaele, R.; Dance, S.L.; Ojha, V. Deep learning for the estimation of water-levels using river cameras. *Hydrol. Earth Syst. Sci. Discuss.* **2021**, *25*, 4435–4453. [\[CrossRef\]](#)

58. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
59. Chen, L.C.; Papandreou, G.; Schroff, F.; Adam, H. Rethinking atrous convolution for semantic image segmentation. *arXiv* **2017**, arXiv:1706.05587.
60. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 2481–2495. [\[CrossRef\]](#)
61. Chen, L.-C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. In Proceedings of the Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 801–818.
62. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected CRFs. *arXiv* **2014**, arXiv:1412.7062.
63. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
64. Feurer, M.; Hutter, F. Hyperparameter Optimization. In *Automated Machine Learning*; Springer Nature: Charm, Switzerland, 2020; pp. 409–411. ISBN 9783030053178.
65. Alex, K.; Ilya, S.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks Alex. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [\[CrossRef\]](#)
66. Csurka, G.; Larlus, D.; Perronnin, F. What is a good evaluation measure for semantic segmentation? In Proceedings of the BMVC 2013—British Machine Vision Conference 2013, Bristol, UK, 9–13 September 2013. [\[CrossRef\]](#)
67. Zhang, Q.; Karkee, M.; Zhang, Q.; Whiting, D. Apple Tree Trunk Trunk and Branch Segmentation Segmentation for for Automatic Automatic Trellis Trellis Training Training Apple Tree Trunk and Segmentation for Automatic Trellis Training Using Convolutional Neural Network Based Semantic Segmentation. *IFAC-PapersOnLine* **2018**, *51*, 75–80. [\[CrossRef\]](#)
68. Martin, D.R.; Fowlkes, C.C.; Malik, J. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. Pattern Anal. Mach. Intell.* **2004**, *26*, 530–549. [\[CrossRef\]](#) [\[PubMed\]](#)
69. Spearman, C. The Proof and Measurement of Association between Two Things. *Am. J. Psychol.* **1904**, *15*, 72–101. [\[CrossRef\]](#)
70. Vandaele, R.; Dance, S.L.; Ojha, V. Automated Water Segmentation and River Level Detection on Camera Images Using Transfer Learning. In *DAGM German Conference on Pattern Recognition*; Springer: Cham, Switzerland, 2021; pp. 232–245. [\[CrossRef\]](#)
71. Boonpook, W.; Tan, Y.; Ye, Y.; Torteeka, P.; Torsri, K.; Dong, S. A deep learning approach on building detection from unmanned aerial vehicle-based images in riverbank monitoring. *Sensors* **2018**, *18*, 3921. [\[CrossRef\]](#) [\[PubMed\]](#)
72. Alqazzaz, S.; Sun, X.; Yang, X.; Nokes, L. Automated brain tumor segmentation on multi-modal MR image using SegNet. *Comput. Vis. Media* **2019**, *5*, 209–219. [\[CrossRef\]](#)
73. Torres, D.L.; Queiroz Feitosa, R.; Nigri Happ, P.; Elena Cué La Rosa, L.; Marcato Junior, J.; Martins, J.; Olã Bressan, P.; Gonçalves, W.N.; Liesenberg, V. Applying Fully Convolutional Architectures for Semantic Segmentation of a Single Tree Species in Urban Environment on High Resolution UAV Optical Imagery. *Sensors* **2020**, *20*, 563. [\[CrossRef\]](#)
74. Talal, M.; Panthakkan, A.; Mukhtar, H.; Mansoor, W.; Almansoori, S.; Ahmad, H. Al Detection of Water-Bodies Using Semantic Segmentation. In Proceedings of the 2018 International Conference on Signal Processing and Information Security (ICSPIS), Dubai, UAE, 7–8 November 2018. [\[CrossRef\]](#)
75. Baheti, B.; Innani, S.; Gajre, S.; Talbar, S. Semantic scene segmentation in unstructured environment with modified DeepLabV3+. *Pattern Recognit. Lett.* **2020**, *138*, 223–229. [\[CrossRef\]](#)
76. Torres, D.L.; Feitosa, R.Q.; La Rosa, L.E.C.; Happ, P.N.; Marcato Junior, J.; Gonçalves, W.N.; Martins, J.; Liesenberg, V. Semantic segmentation of endangered tree species in Brazilian savanna using DEEPLABV3+ variants. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.-ISPRS Arch.* **2020**, *42*, 355–360. [\[CrossRef\]](#)
77. Garcia-Garcia, A.; Orts-Escolano, S.; Oprea, S.; Villena-Martinez, V.; Garcia-Rodriguez, J. A Review on Deep Learning Techniques Applied to Semantic Segmentation. *arXiv* **2017**, arXiv:1704.06857.
78. Pashaei, M.; Kamangir, H.; Starek, M.J.; Tissot, P. Review and evaluation of deep learning architectures for efficient land cover mapping with UAS hyper-spatial imagery: A case study over a wetland. *Remote Sens.* **2020**, *12*, 959. [\[CrossRef\]](#)
79. Li, Z.; Wang, R.; Zhang, W.; Hu, F.; Meng, L. Multiscale features supported deeplabv3+ optimization scheme for accurate water semantic segmentation. *IEEE Access* **2019**, *7*, 155787–155804. [\[CrossRef\]](#)
80. Lo, S.W.; Wu, J.H.; Chen, L.C.; Tseng, C.H.; Lin, F.P.; Hsu, C.H. Uncertainty comparison of visual sensing in adverse weather conditions. *Sensors* **2016**, *16*, 1125. [\[CrossRef\]](#) [\[PubMed\]](#)
81. Hsu, S.-Y.Y.; Chen, T.-B.B.; Du, W.-C.C.; Wu, J.-H.H.; Chen, S.-C.C. Integrate Weather radar and monitoring devices for urban flooding surveillance. *Sensors* **2019**, *19*, 825. [\[CrossRef\]](#)