remote sensing

MDPI

Article

A Real-Time Detecting Method for Continuous Urban Flood Scenarios Based on Computer Vision on Block Scale

Haocheng Huang [1,2], Xiaohui Lei [2], Weihong Liao [2], Haichen Li [2,*], Chao Wang [2] and Hao Wang [2]

[1] School of Civil Engineering, Central South University, Changsha 410075, China
[2] State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, China Institute of Water Resources and Hydropower Research, Beijing 100038, China; lxh@iwhr.com (X.L.)
* Correspondence: li0haichen@foxmail.com

Abstract: Due to the frequent and sudden occurrence of urban waterlogging, targeted and rapid risk monitoring is extremely important for urban management. To improve the efficiency and accuracy of urban waterlogging monitoring, a real-time determination method of urban waterlogging based on computer vision technology was proposed in this study. First, city images were collected and then identified using the ResNet algorithm to determine whether a waterlogging risk existed in the images. Subsequently, the recognition accuracy was improved by image augmentation and the introduction of an attention mechanism (SE-ResNet). The experimental results showed that the waterlogging recognition rate reached 99.50%. In addition, according to the actual water accumulation process, real-time images of the waterlogging area were obtained, and a threshold method using the inverse weight of the time interval (T-IWT) was proposed to determine the times of the waterlogging occurrences from the continuous images. The results showed that the time error of the waterlogging identification was within 30 s. This study provides an effective method for identifying urban waterlogging risks in real-time.

Keywords: urban waterlogging; real-time monitoring; computer vision; ResNet

Citation: Huang, H.; Lei, X.; Liao, W.; Li, H.; Wang, C.; Wang, H. A Real-Time Detecting Method for Continuous Urban Flood Scenarios Based on Computer Vision on Block Scale. *Remote Sens.* **2023**, *15*, 1696. https://doi.org/10.3390/rs15061696

Academic Editor: Tiziana D'Orazio

Received: 29 January 2023
Revised: 14 March 2023
Accepted: 15 March 2023
Published: 21 March 2023

1. Introduction

Conventional urban waterlogging monitoring systems are comprised of water-level monitoring equipment installed at specific locations to obtain waterlogging depths and determine whether the water level exceeds the warning water level. This equipment includes devices such as pressure-sensing water level gauges and ultrasonic sensors. However, in the management of real projects, these water-level monitoring devices are often installed in drainage wells which are easily damaged by the scouring of rainwater and sewage, as well as vibrations from passing vehicles. The low utilization rate of the monitoring equipment and the difficulty of determining the status of the equipment makes it difficult to inspect the water-level gauge daily. Rather than using water-level gauges that can only obtain data from a single location, real-time images and videos can effectively visually monitor urban waterlogging. Although numerous municipal surveillance cameras are installed in cities and mobile phones offer cost-friendly and increasingly accessible photographs, it is labor-intensive to manually analyze a large number of street photographs and videos.

With the increase in computing power, computer vision technology [1] has developed rapidly and has been widely applied in fields such as biomedicine [2–5], autonomous driving [6–8], commercial systems [9,10], and agriculture [11]. In recent years, computer vision has also been applied to the prophylaxis and treatment of COVID-19 [12,13]. These technologies have also been promoted in hydraulic research and urban management. Remote sensing image scene classification, land use, and land cover changes [14–17] based on computer vision methods with convolutional neural network (CNN) algorithms have proven to be effective. In addition, image-based algorithms are also used in flood

monitoring tasks, such as detecting flooded areas of river channels using different deep learning image segmentation neural networks [18,19] and observing water levels with deep learning-based unmanned surveillance.

With the frequent occurrence of urban waterlogging, tentative research on image-based urban flood monitoring has been carried out, such as a method combining traditional monitoring with video image estimations of flood areas or water levels [20–22], which used the single-shot MultiBox detector target detection algorithm to detect the shape of the traffic bucket in a flood scenario and determine the water depth [23]. In addition, many studies on flood recognition and segmentation are based on satellite images, for example, U-Net is used for flood semantic classification [24]. Further, many studies are based on discrete images or real-time traffic camera photos [25,26], which are of great significance. However, most of the previous studies have focused on single-image recognition without considering the continuity of the flooding process and the real-time requirements of recognition. Therefore, it is still an urgent task to propose a general method for monitoring urban waterlogging based on computer vision in flooding emergency sites.

This study explored the use of computer vision algorithms to identify urban waterlogging risks through images captured by the public and by monitoring equipment. In Section 2, a method of flooding detection based on computer vision using FLI is proposed. In Section 3, two cases for flooding detection with photos from the public and a fixed camera on a block, respectively, are discussed. In Section 4, the results and contributions of our method are discussed. Finally, a summary is presented.

2. Methodology

2.1. Computer Vision

The main tasks of image processing with computer vision technology include feature extraction, segmentation [27], classification [28], recognition [29], and detection [30]. Overall, these technologies aim to bridge the semantic gap.

Traditional image classification uses a rule-based approach to identify object categories by hard coding, that is, extracting explicit rules and organizing the human understanding of objects into codes. Two methods typically fall into this category. The first method is the bag of words model [31], which uses edge algorithms, such as the Canny algorithm, to obtain object information and extract local feature regions to describe objects (such as SIFT), and then it uses clustering algorithms (such as K-means) to select the word in the bag to describe the image. This method can reduce the dimensions of image data while avoiding the problem of algorithm failure caused by unfavorable factors, such as the occlusion of feature points in the global feature method. This approach is effective for recognizing object categories with relatively stable features. The second method is to represent the image according to global features [32], that is, to extract frequency features from an image. This method divides an image into blocks, represents the frequency of each block, and then uses the frequency as a feature vector to describe the image. The method is suitable for large scenes such as landscapes and urban buildings. Despite the advantages of these traditional image classification methods, they still encounter difficulties, including perspective, illumination, scale, occlusion, deformation, background clutter, intra-class deformation, and motion blur. Object recognition is difficult for objects with rich morphological changes, which are greatly affected by changes in perspective scale, especially objects with severe deformation.

Unlike these traditional methods, machine-learning-based computer vision methods have proven to be highly effective in the image classification and detection of irregular objects [33].

2.1.1. Classification Model Based on Deep Learning

As a waterlogging incident has an unstable shape, we considered using deep-learning-based computer vision algorithms to identify urban waterlogging statuses. The steps of the image classification method based on deep learning include data set construction, classifier

design and learning, and classifier decision. The classifier design and decision-making processes are illustrated in Figure 1.

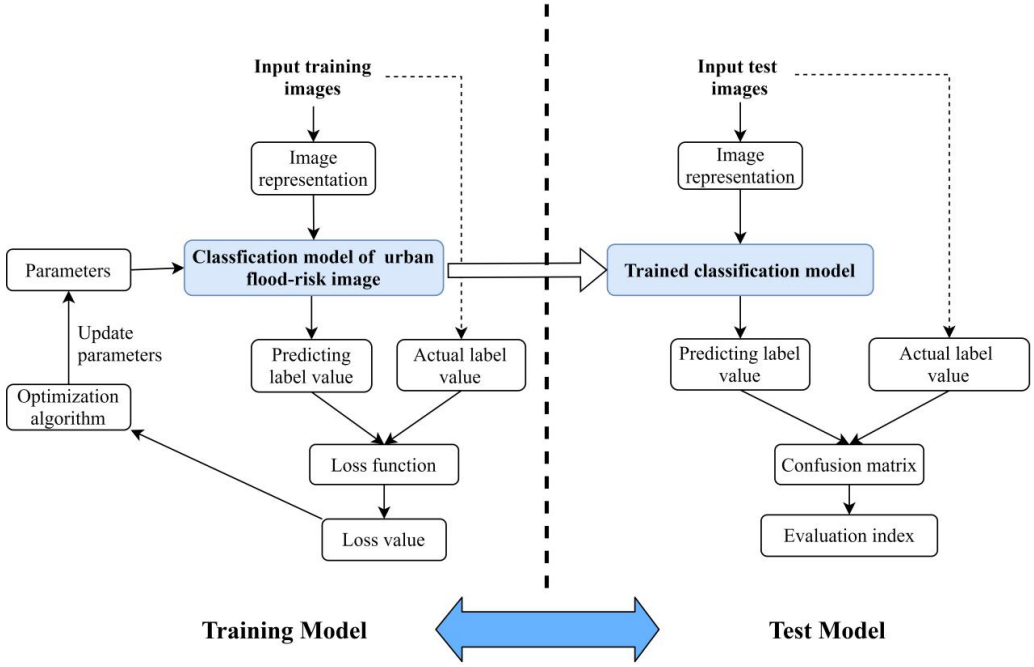

Figure 1. Processes of classifier design and decision-making.

As most classification algorithms require vectors as an input, images must first be converted to vectors. An image can be converted into a vector in two ways: feature-based representation and pixel-based representation. Feature-based representation mainly includes global features and local features, whereas pixel-based representation converts the RGB value corresponding to each pixel into a vector. Pixel-based representation involves a high number of dimensions. However, algorithm optimization and computing power improvements have gradually helped to address this problem.

Commonly used models include nearest neighbor models (such as K-nearest neighbor), Bayesian, linear, support vector machine, neural network, random forest, and AdaBoost models. In addition, certain models exist in special application environments, such as SqueezeNet for limited storage resources and MobileNet and ShuffleNet [34,35] for limited computing resources. Optimization algorithms mainly include first-order algorithms (gradient descent, stochastic gradient descent, and mini-batch stochastic gradient descent) and second-order algorithms (Newton's method, BFGS, and L-BFGS). Common evaluation indicators include accuracy, error rate, top 1 indicator, and top 5 indicators. Classic neural network models include AlexNet, ZFNet, VGG, GoogLeNet, ResNet, and Inception, and ResNet and Inception V4 have the best generalization performance. Therefore, the ResNet model was used in this study for waterlogging image recognition.

2.1.2. Models

1. ResNet

Typically, as a network layer increases, the gradient of the network disappears or decreases, increasing the error rate. This phenomenon is called network degradation. ResNet was first proposed by He [36] and addresses the network degradation problem by introducing a deep residual learning framework with shortcut connections (Figure 2). The algorithm has proven to be highly effective for image classification and image segmentation tasks [37].

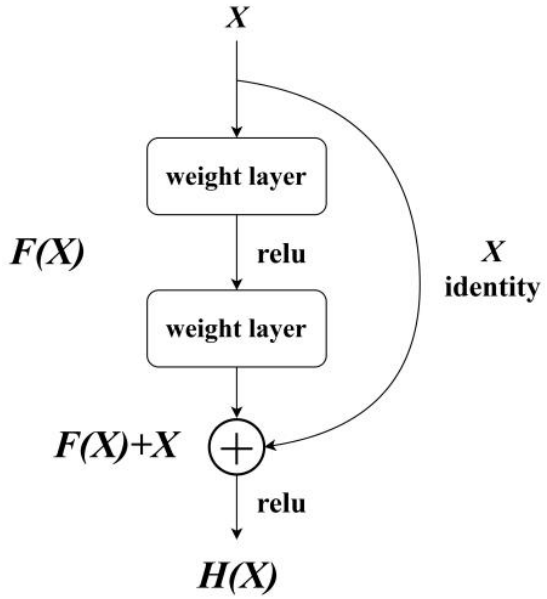

Figure 2. Shortcut connections.

The residual structure establishes a mapping relationship:

$$H(X) = F(X) + X \tag{1}$$

where X is the input network, $F(X)$ is the current transformation, and $H(X)$ is the output network. By introducing the residual structure (shortcut connections), the ResNet algorithm retains the key information of the previous network layer, enhances the feature information of interest, avoids the problem of deep network reduction, and solves the performance degradation problem caused by the deep network.

One explanation for why residual networks perform well in image classification tasks is that residual networks can be regarded as an integrated model consisting of many sub-networks (Figure 3).

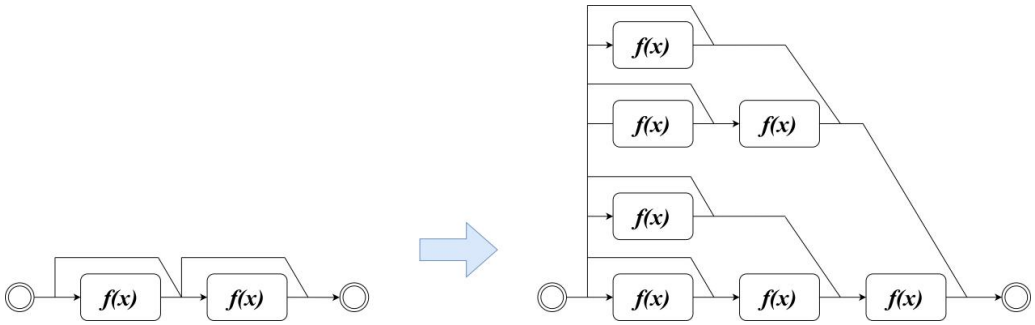

Figure 3. Expansion of the residual structure.

2. EfficientNet

The core of EfficientNet is to construct a standardized convolutional network expansion method [38] that can achieve high accuracy and save computational resources, that is, to optimize the efficiency and accuracy of the network by balancing the three dimensions of resolution, depth, and width (Figure 4). As seen in Figure 4a is EfficientNet's baseline and (b) is the main idea of EfficientNet, which is to comprehensively expand the width, depth, and resolution of a network.

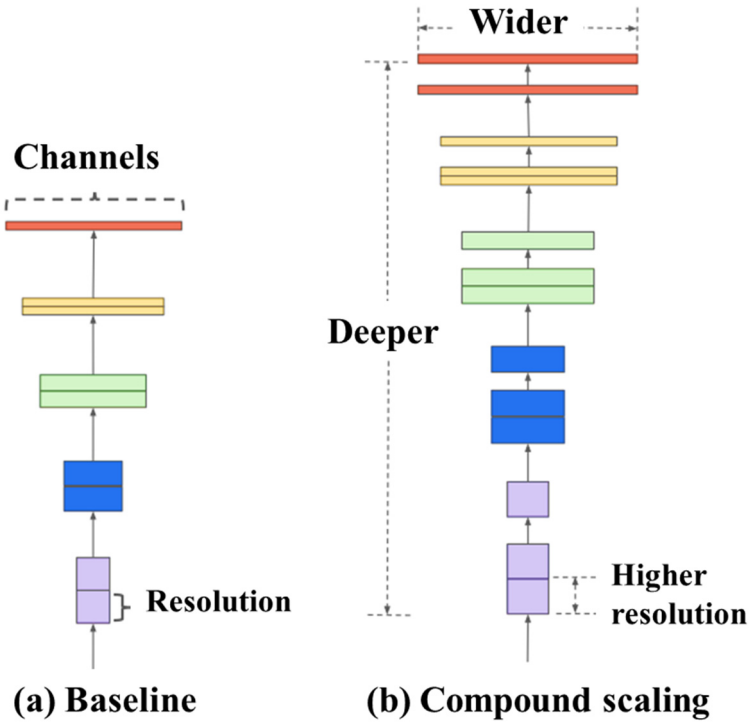

Figure 4. EfficientNet.

3. 3D CNN

A 3D CNN is used for human action recognition [39] based on video classification. Image classification based on a 2D CNN only considers the feature of a single image, while a 3D CNN considers the dynamic relationship between consecutive images to capture the motion information between frames (Figure 5). The value at position (x, y, and z) on *j*th feature map in the *i*th layer is given by:

$$v_{ij}^{xyz} = \tanh\left(b_{ij} + \sum_{m}\sum_{p=0}^{P_i-1}\sum_{q=0}^{Q_i-1}\sum_{r=0}^{R_i=1}\omega_{ijm}^{pqr}v_{(i-1)m}^{(x+p)(y+q)(z+r)}\right) \quad (2)$$

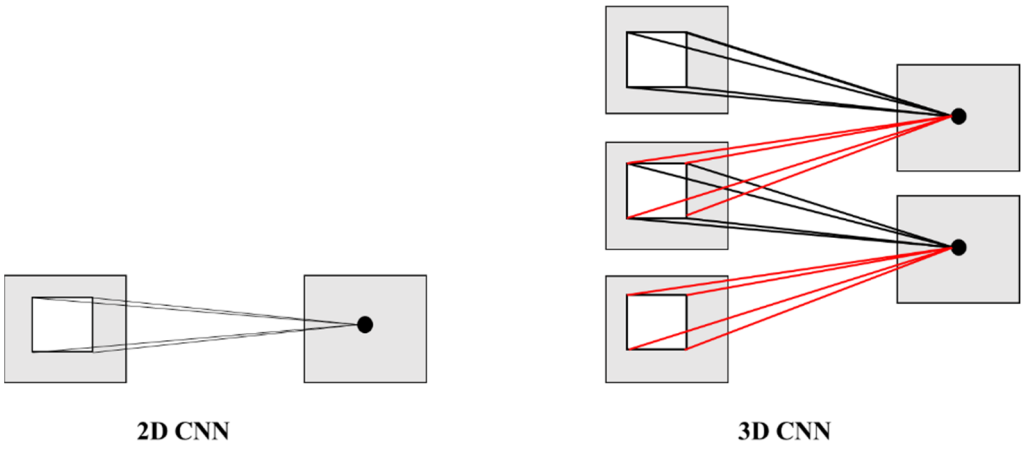

Figure 5. Standard 2D and 3D CNNs.

4. Other models

MobileNet is based on using depth-wise separable convolutions to reduce a model's volume and speed up computation [34], which has proven to be efficient on mobile phones. DenseNet [40] provides a dense connection that connects each layer to every other layer in a feed-forward fashion. ViT shows that a pure transformer applied directly to sequences of image patches performs well for image classification tasks.

Different models are applicable to different tasks and different types of hardware [17]. Considering that urban waterlogging identification is mainly carried out at mobile workstations, ResNet, EfficientNet, and a 3D CNN were used as the models in this study.

2.1.3. Attention

Attention mechanisms [41] originated from the study of the principles of human vision. The human visual system generates different acuities for different pieces of information in an image. The human visual system efficiently uses visual processing resources by focusing on important areas and ignoring unimportant information.

In this study, squeeze-excitation (an SE-block) was used to build a channel attention model that explicitly modeled the interdependencies between the feature channels. Specifically, the importance of each feature channel was automatically obtained through learning. Then, according to the degree of importance, the network used global information to selectively enhance the beneficial feature channels and suppress the useless feature channels. Adaptive feature channel calibration was thereby achieved.

The structure of ResNet with an SE-block (SE-ResNet) is illustrated in Figure 6.

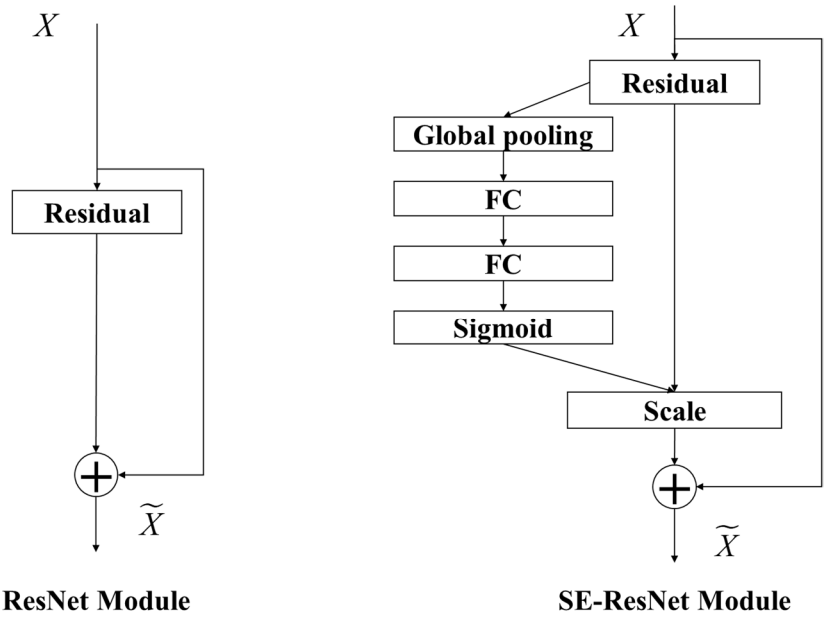

Figure 6. SE-ResNet.

2.1.4. Data Augmentation

The simulation performance of deep neural networks directly depends on the dataset quantity and quality. In general, the larger the scale and the higher the quality of the dataset, the higher the generalization of the model. Therefore, a large amount of training data is required to ensure a good model performance to obtain ideal results. However, in practical engineering, the collected data seldom cover all scenarios. For example, different pictures of a particular scene may have various picture effects due to different lighting conditions. Hence, when training the model, data augmentation in lighting is required. Data augmentation [42] is an important approach to increasing the diversity of training samples

and enlarging datasets, and it helps to increase the amount of relevant data, improve model robustness, prevent the model from learning undesired models, and avoid overfitting.

Common and effective image data augmentation methods include flipping, shifting, rotating, random cropping, color jittering, and random noise introduction. These approaches are suitable for smaller datasets and can effectively increase the number of images.

In addition, the early stop method is also an important means for solving the overfitting problem. It is a callback to specifics where parameters such as accuracy or loss should be judged by rules at the beginning and end of each epoch to determine whether the model should be stopped.

2.2. Threshold Method of the Time Interval Inverse Weight

Compared with the waterlogging features in published photos, an urban waterlogging process detected by an urban camera monitoring system is continuous and gradual, leading to extremely complex factors affecting the determination results in actual scenarios. When rainfall reaches a certain level, the real scene transitions from non-flooding to flooding. The result may be unreliable during this period, regardless of whether the image is recognized manually or through a computer vision algorithm.

To solve this problem, we proposed a threshold method of the inverse weight of the time interval (T-IWT). This method converts the problem of determining whether the image is negative (non-flooding) or positive (flooding) at each moment into seeking the critical time closest to the actual waterlogging occurrence.

By analyzing and scoring the judgments of the previous moments, the flood likelihood index (FLI) is obtained. The first time at which the FLI exceeds the threshold is regarded as the critical time when the waterlogging occurs.

The FLI is defined as:

$$FLI = \frac{\sum_{i=1}^{n} S_{t-i}\lambda_{t-i}}{\sum_{i=1}^{n} \lambda_{t-i}} \tag{3}$$

The inverse interval weight (IIW) is calculated as follows:

$$\lambda_{t-i} = \frac{1}{i} \tag{4}$$

The inverse average weight (IAW) is calculated as follows:

$$\lambda_{t-i} = \frac{1}{n} \tag{5}$$

The inverse time-step weight (ITW) is calculated as follows:

$$\lambda_{t-i} = \frac{1}{\left\lfloor \frac{i}{m} \right\rfloor + 1} \tag{6}$$

For continuous scenes, the time error ε is used as the evaluation index of recognition accuracy, as follows:

$$\varepsilon = t_A - t_I \tag{7}$$

Descriptions of the parameters are summarized in Table 1.

Table 1. Descriptions of the parameters.

Parameter	Description
S	The model judgment result (no waterlogging risk/negative is recorded as 0, and the waterlogging risk/active is recorded as 1)
n	The backtracking time
t	The time of determination
λ	The weight
m	The number of image frames in a unit time interval
$\left\lfloor \frac{i}{m} \right\rfloor$	The rounded-down result of $\frac{i}{m}$
t_A	The actual time of waterlogging
t_I	The time of the waterlogging occurrence obtained by the threshold method of the inverse weight of the time interval

3. Case Study

3.1. Case 1: Waterlogging Recognition for Public Image Data

The dataset for this case comprised publicly available images. Through Google, Baidu, and other search engines, keywords such as "waterlogging" and "flooding" were searched, and pictures of waterlogging on public websites were obtained. Non-flooding photographs of cities were obtained by using search keywords such as "street" and "boulevard." The dataset in this case was formed through manual screening.

Overall, the dataset comprised 2245 (Figures 7–9) street-view photographs of cities. There were 1107 photographs without floods, which were labeled as Category 0, and there were 1118 photographs with floods, which were labeled as Category 1.

Figure 7. Images of urban streets without floods.

Figure 8. Images of urban streets with floods.

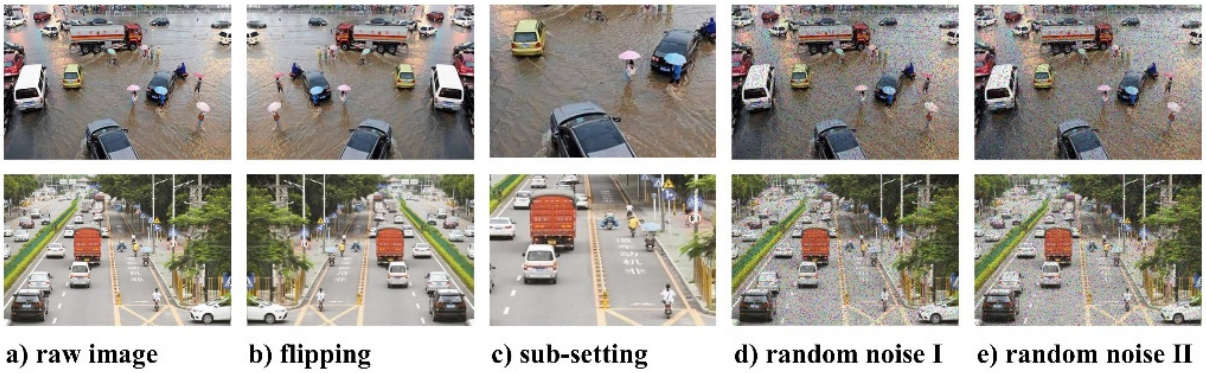

Figure 9. Data augmentation.

In these datasets, 298 images without floods and 298 images with floods were set as the validation datasets. The remaining 809 photos without floods and 820 photos with floods were used as the training datasets.

In addition, to improve the accuracy of the model, data enhancement methods, such as flip, sub-set, and random noise addition, were used to increase the number of training sets. By augmenting the dataset, 9802 training-set photographs were obtained, including 4771 photographs without floods and 5031 with floods.

3.2. Case 2: Waterlogging Recognition for Actual Scenarios

3.2.1. Experimental Setup

Fuzhou, the capital of Fujian Province, is located on the southeastern coast of China (26°08′N, 119°28′E). In the experiment, a camera was installed at an overpass above (Figure 10) the road and aimed at the overflowing municipal pipe network inspection well to obtain a video stream. The picture resolution was 2560 × 1440, with 96 dots per inch, and the frame rate of the camera was 25 frames per second.

Figure 10. The experimental setup.

The camera captured video footage from 4:00 on 17 May 2021 to 12:00 on 19 May 2021. During this period, from 20:00 to 22:44 on 17 May and from 16:58 on 18 May to 5:04 on 19

May, the actual scene had flooding. The video stream from 4:00 to 10:00 on 17 May was used as the data source of the non-flooding samples in the training dataset, and the video streams from 20:30 to 22:30 on 17 May and from 0:00 to 4:00 on 19 May were used as data sources for the flooding samples in the training dataset. The video stream from 15:00 to 18:30 on 18 May was used as the data source of the validation dataset, and the moment at 16:58 was used as the critical time to distinguish between the scenes with or without floods.

The video stream was captured at a frequency of one frame every 5 s. A total of 8640 training dataset images were obtained, and 4320 images were labeled as with or without floods. A total of 2520 validation dataset images were obtained, and 1416 of these were labeled as without floods and 1104 were labeled as with floods.

The SE-ResNet model was trained to perform this task.

3.2.2. Training Model

To determine whether an image depicts flooding, an image recognition algorithm is conventionally used to determine whether an image obtained for a particular moment is under flood risk. However, this method does not produce ideal results in actual recognition work. The simulation results of this study showed that the scene recognition accuracy did not achieve the expected effect for image classification at a single moment (Figure 11), and the overall recognition accuracy was 91.6%. The accuracy of the model was low, especially for the period before the waterlogging occurred. Meanwhile, the recognition results of the model showed irregular positive results. However, in a real waterlogging scene, the process from no waterlogging to waterlogging is transitional, and the sudden appearance and disappearance of waterlogging is rare.

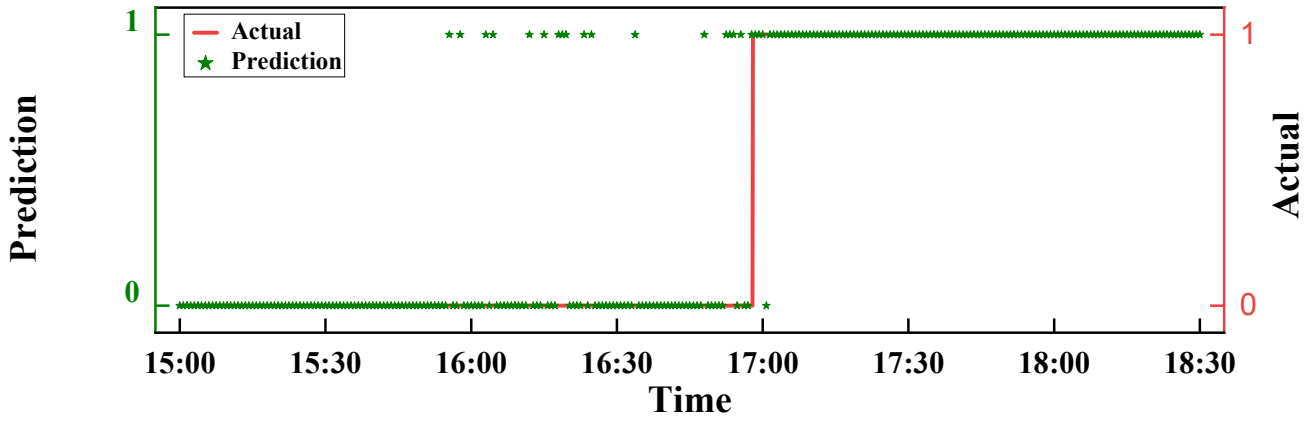

Figure 11. Result of image recognition using SE-ResNet: 1 denotes a positive recognition result (a risk of waterlogging exists) and 0 denotes a negative recognition result (no risk of waterlogging exists).

Further analysis of the distribution characteristics of the research validation set prediction results showed that when the degree of waterlogging in the prediction set image was extremely low (15:00–15:45) or high (17:15–18:30), the recognition accuracy of the model reached 100%. Therefore, it is crucial to determine a reasonable critical time for distinguishing waterlogging situations in actual scenarios.

4. Results and Discussion

4.1. Results of Case 1

Figure 12 shows the accuracy and loss curve of each model with increasing numbers of epochs. The results showed that the model performance increased with the increasing numbers of training epochs and datasets.

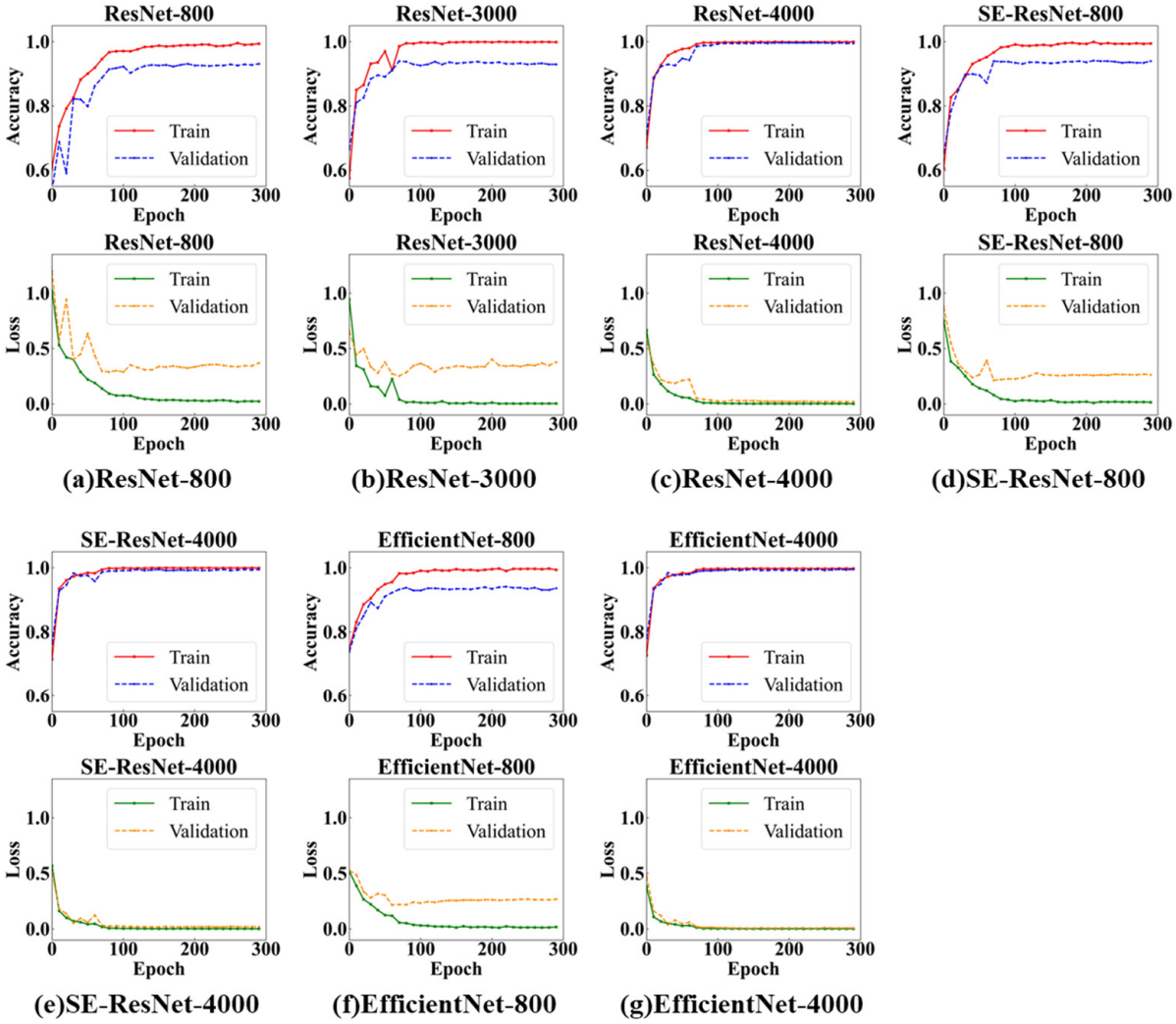

Figure 12. Simulation results of each model. The number after the model name indicates the number of training datasets.

When the number of training datasets was 800, the recognition accuracy of both the ResNet and SE-ResNet models on the validation datasets reached 93%, and no significant difference existed between the models in the index of the highest accuracy. Meanwhile, Figure 13 shows that the numbers of epochs were 6, 22, and 71 when ResNet reached 70%, 80%, and 90% accuracy, respectively, and the numbers of epochs were 1, 9, and 41 when SE-ResNet reached corresponding accuracies, respectively. Thus, compared with ResNet, SE-ResNet achieved higher accuracy with fewer training epochs. Similar to SE-Resnet, EfficientNet achieved higher accuracy faster than ResNet on fewer data sets, but the highest accuracy was still limited by the amount of data. With 800 training datasets, EfficientNet achieved 80% and 90% levels of accuracy at the 9th and 36th epochs, respectively, which was close to half of the number of training rounds required by ResNet, but the best accuracy was not significantly improved. With 4000 training datasets, EfficientNet could also achieve more than 99.5% accuracy, which was close to the accuracy achieved by SE-ResNet (Table 2).

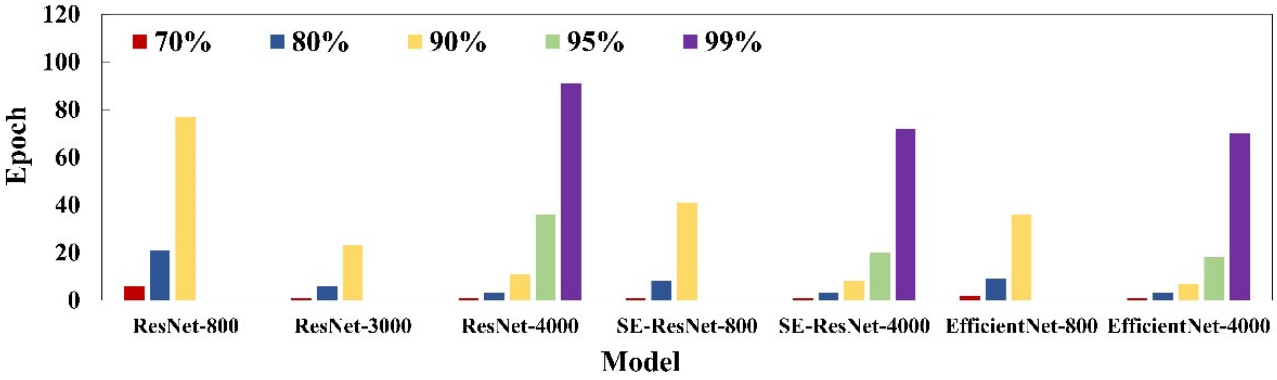

Figure 13. Number of epochs when the model result reached target accuracy.

Table 2. Highest recognition accuracy of each model and the corresponding number of epochs.

	Highest Accuracy	Best Number of Epochs
ResNet-800	93.46%	286
ResNet-3000	94.30%	72
ResNet-4000	99.83%	201
SE-ResNet-800	94.30%	214
SE-ResNet-4000	99.50%	83
Efficient-800	94.52%	208
Efficient-4000	99.52%	70

As the amount of training data increased, the recognition accuracy of the model gradually increased, and the number of epochs that achieved the same recognition rate gradually decreased. When the number of training datasets reached the order of 4000, the accuracy of the model for both the training and validation datasets stably reached 99.5%.

Therefore, the following conclusions were drawn. First, models that introduced an attention mechanism could achieve higher accuracy with a smaller number of training epochs. Second, the number of training datasets determined the upper limit of the recognition accuracy of the model.

4.2. Results of Case 2

Figure 14 shows the waterlogging process recorded by a camera and the recognition results of the models. At 16:58, obvious waterlogging appeared in the image. At 16:56, the model incorrectly judged the result to be positive. However, using the threshold method of the inverse weight of the time interval, the recognition results were delayed to varying degrees compared with the actual time.

The results (Figure 15) showed that the error was negatively correlated with the threshold and backtracking time. The judgment errors of all modes were greater than 2000s when the backtracking time was less than 1 min or the threshold was less than 0.7, and when the threshold was greater than 0.9 and the backtracking time was greater than 3 min, the errors were all less than 200 s. A possible cause of this is that there were local dense positive judgment results based on the SE-ResNet model around a certain time period, and the current time results were assigned an excessively high weight when the backtracking time was set to be too short, and the low threshold made it easier for the *FLI* to reach criticality.

Figure 14. Waterlogging images in real scenes. P denotes a positive result and N denotes a negative result.

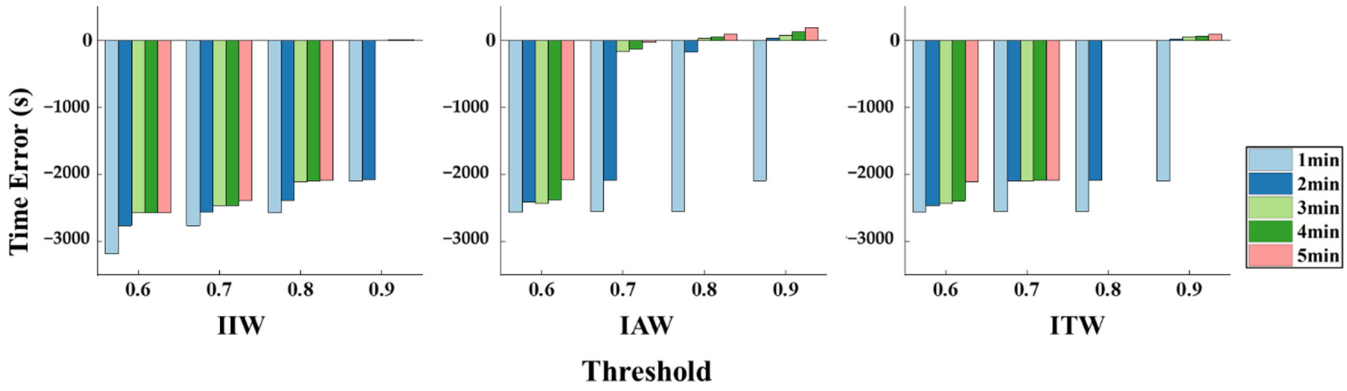

Figure 15. Errors corresponding to the *FLI* obtained by the different methods.

In addition, compared with the IIW, the backtracking time constraints of the IAW and ITW were less rigorous. For the IIW, the error time was less than 10 s only when the backtracking time was greater than 2 min and the threshold was set to 0.9, which was more than 30 min under the other threshold and backtracking time settings. On the other hand, it was indicated that the response to waterlogging identification was rapid when the parameters of the IIW were reasonably set.

However, when the error was less than 5 min, the result of the IAW (Figures 16 and 17) tended to be more delayed than that of the ITW. For example, when the threshold was set to 0.8 and the backtracking time was set to 3 min, the error of the IAW was 25 s, while that of the ITW was -5 s. It is worth noting that when the threshold was 0.9 and the backtracking time was 2 min, the time errors of both methods were less than 30 s (25 s and 15 s, respectively).

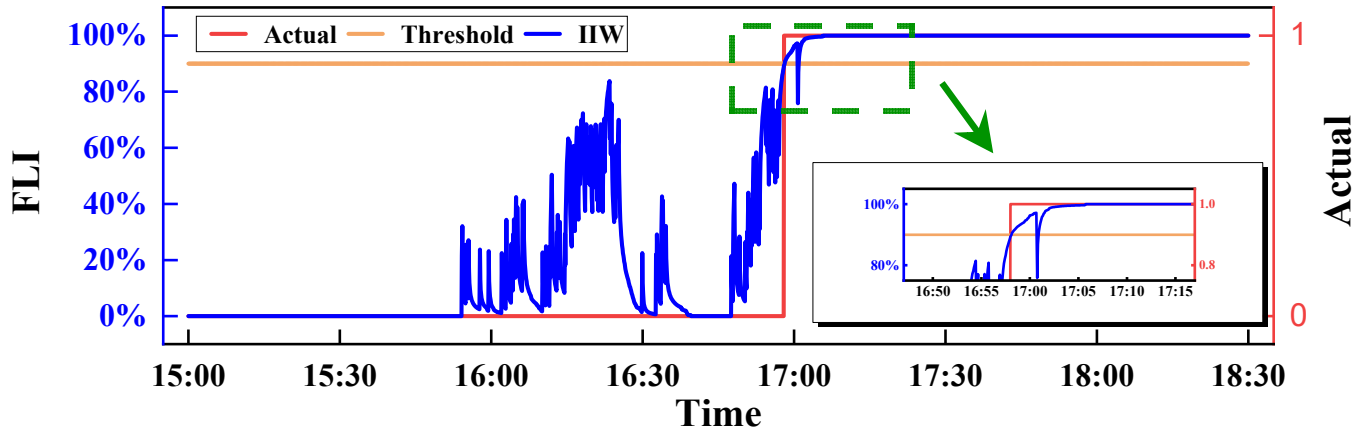

Figure 16. The *FLI* curve with the IIW.

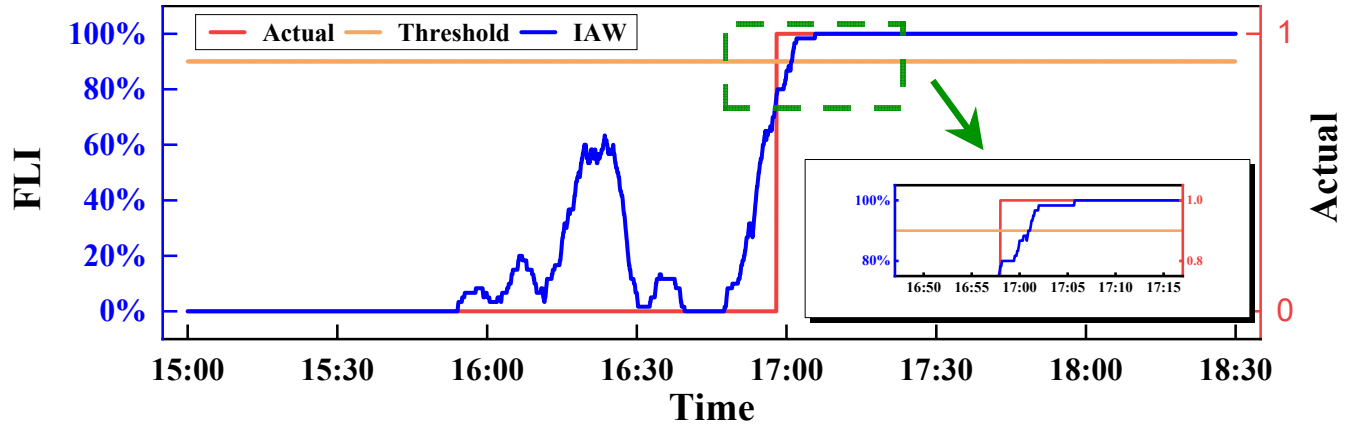

Figure 17. The *FLI* curve with the IAW.

For example, as shown in Figure 16, when the IIW was used as the evaluation basis for waterlogging recognition, the *FLI* curve oscillated sharply during the waterlogging process. This phenomenon occurred because the evaluation weight of the IIW on the image recognition results near each moment was relatively large, and the sensitivity to the recognition results of the waterlogging scenes was relatively strong. The evaluation weight of the single preceding moment accounted for 21.36% of the total evaluation weight, whereas the evaluation weight of the previous 12 moments (1 min) accounted for 52.35%. Hence, after rainfall, the index frequently appeared in 60% to 80% of the results and quickly decreased to below 50%. In addition, at 17:02, the index decreased once to 80%, which was lower than the threshold, and the scene was mistakenly determined as a non-flooding scene. Therefore, the use of the IIW for waterlogging recognition has the shortcomings of instability and low accuracy. However, the IIW method responds faster to flooding scenarios, with nearly no time error.

When the IAW was used as the evaluation basis for waterlogging identification, as shown in Figure 15, the *FLI* curve exhibited a gentle trend during the waterlogging process. The IAW evenly assigned weights to all images within the first 5 min; hence, the images within the discriminant range showed no significant differences in the impacts of the recognition results. Before the occurrence of waterlogging, although some images were positive, the *FLI* was always lower than 75%, and after the occurrence of waterlogging, some images were negative but no obvious decrease or shock occurred. However, the IAW method showed a higher delay in recognizing the occurrence of waterlogging, and the time error of the recognition was 190 s later than the actual time.

As the evaluation basis for waterlogging identification, the ITW simultaneously considered the stability and timeliness of the judgment. As shown in Figure 18, the *FLI* curve with the ITW was relatively flat. The ITW divided the images into five levels at intervals of 1 min and assigned weights that were positively correlated with the time interval. Before the occurrence of waterlogging, the *FLI* was lower than 80%, and after the occurrence of waterlogging, although some images were judged as negative, the *FLI* only fluctuated slightly and did not fall below the threshold. Moreover, the delay in the ITW method in recognizing the occurrence of waterlogging scenes was within an acceptable range, and the time error was only 90 s later than the actual waterlogging occurrence time.

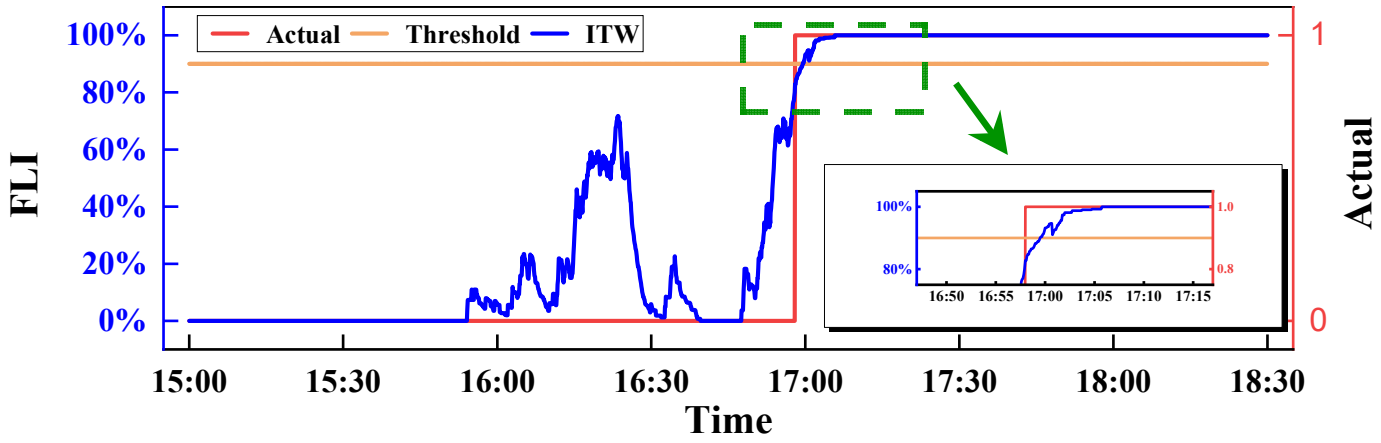

Figure 18. The *FLI* curve with the ITW.

However, the following unresolved issues should be noted in future research. On the one hand, the impact of adverse conditions such as camera vibration and changing light conditions on detection accuracy should be studied. On the other hand, the flood disaster of a single block has been studied, and the widely applicable detection method is crucial but not yet resolved.

In addition, a 3D CNN was used for comparison (Figure 19), referring to a dataset (UCF101) and previous case study [43]. The video training dataset included 180 videos with flooding water and 176 videos without flooding water. The duration of each video was 10 s and the number of training epochs was 150. The results showed that the accuracy of the waterlogging video classification based on the 3D CNN of the training set was only 67.3%. The reason for the unsatisfactory results may have been that the dynamic change in the water was not significant (approximately 3% of the pixels changed every 10 min), while the dynamic change in the other non-target objects (such as vehicles and pedestrians) was too violent. The model was more inclined to capture the dominant dynamic change and could not pay attention to the dynamic characteristics of the research object (water). Therefore, it will be an important task to improve the recognition accuracy of the 3D CNN model for objects with unclear dynamic characteristics under a complex and dynamic background in the future.

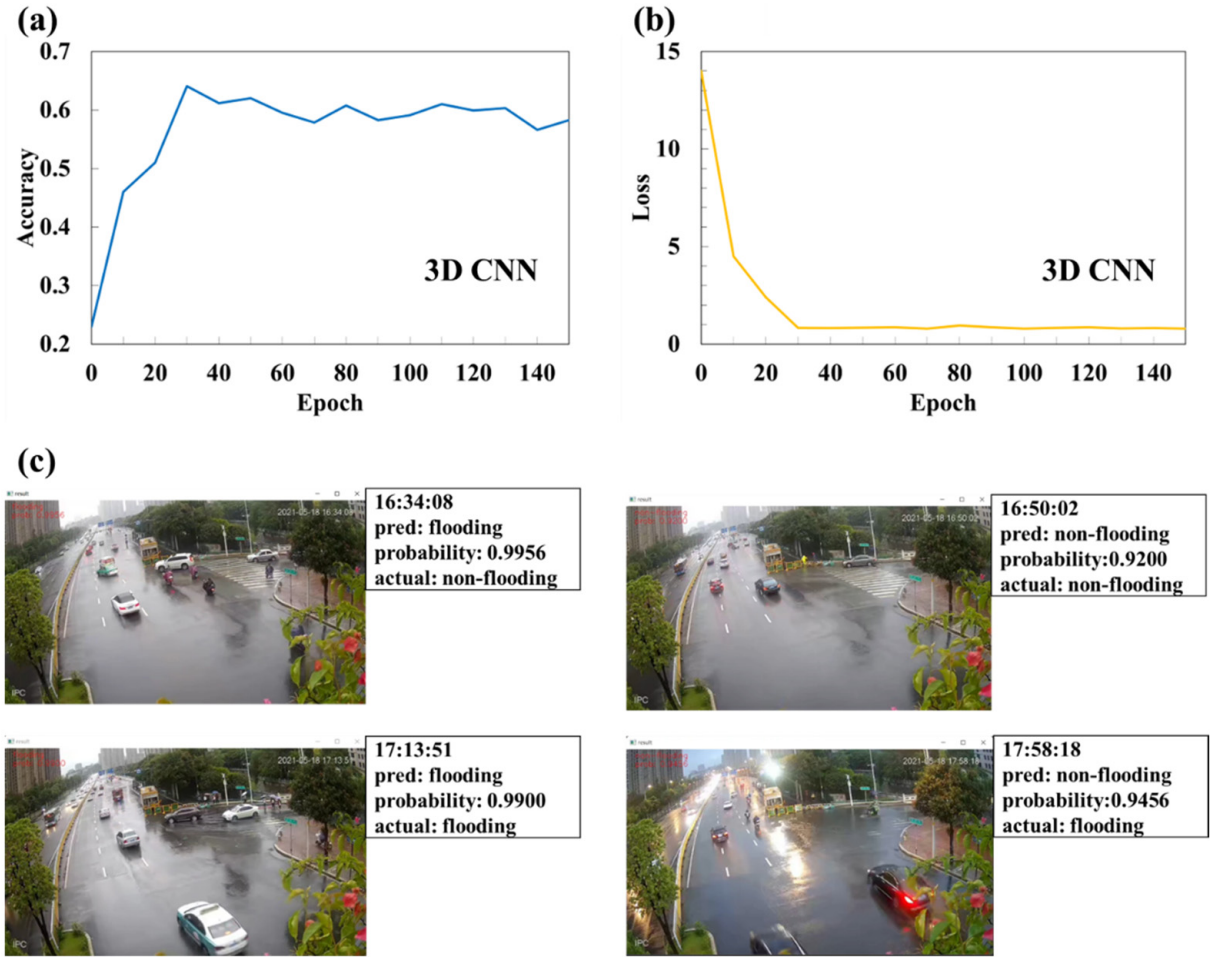

Figure 19. Video classification using the 3D CNN.

5. Conclusions

This study aimed to address the problems of unclear identification and perception of urban waterlogging. Hence, a waterlogging identification method based on computer vision technology was proposed in this study. Waterlogged and non-waterlogged images were used as the input dataset, and a deep neural network (a ResNet model) was built and trained. An attention mechanism was introduced into the model while the data were augmented, improving the model's accuracy. In terms of actual waterlogging scenes, a threshold method of the inverse weight of the time interval (T-IWT) was proposed to determine the occurrence time of waterlogging. The main conclusions of this study are as follows:

1. For the task of waterlogging identification in public image datasets, data augmentation can effectively improve the model's recognition accuracy. When the number of training datasets reaches 4000, the model's accuracy can be stabilized to more than 99%.
2. Compared with the ResNet model, the SE-ResNet model with an attention mechanism achieves higher recognition accuracy with a smaller number of training epochs.
3. For the actual waterlogging scene recognition task, the T-IWT method can effectively achieve waterlogging recognition. Among the flood-likelihood-index definition methods, the inverse average weight (IAW) method and the inverse time-step weight (ITW) method can achieve stable identification, with a model identification response time control falling within 30 s.

Author Contributions: Conceptualization, H.H.; Methodology, H.H.; Software, H.H.; Validation, H.H.; Formal analysis, W.L.; Investigation, X.L.; Resources, X.L. and H.W.; Data curation, H.L. and C.W.; Writing—original draft, H.H.; Writing—review & editing, X.L.; Visualization, X.L.; Supervision, X.L.; Funding acquisition, X.L., H.L. and C.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Key Program of National Natural Science Foundation of China and the National Key R&D Program of China (2022YFC3800102, U2240203, and 2021YFC3001405).

Data Availability Statement: Not applicable.

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

References

1. LeCun, Y.; Kavukcuoglu, K.; Farabet, C. Convolutional Networks and Applications in Vision. In Proceedings of the 2010 IEEE International Symposium on Circuits and Systems, Paris, France, 30 May–2 June 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 253–256.
2. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, 5–9 October 2015; pp. 234–241.
3. Goecks, J.; Jalili, V.; Heiser, L.M.; Gray, J.W. Perspective How Machine Learning Will Transform Biomedicine. *Cell* **2020**, *181*, 92–101. [CrossRef]
4. Galan, E.A.; Zhao, H.; Wang, X.; Dai, Q.; Huck, W.T.S. Review Intelligent Microfluidics: The Convergence of Machine Learning and Microfluidics in Materials Science and Biomedicine. *Matter* **2020**, *3*, 1893–1922. [CrossRef]
5. Litjens, G.; Kooi, T.; Bejnordi, B.E.; Setio, A.A.A.; Ciompi, F.; Ghafoorian, M.; van der Laak, J.A.W.M.; van Ginneken, B.; Sánchez, C.I. A survey on deep learning in medical image analysis. *Med. Image Anal.* **2017**, *42*, 60–88. [CrossRef] [PubMed]
6. Boloor, A.; Garimella, K.; He, X.; Gill, C.; Vorobeychik, Y.; Zhang, X. Attacking vision-based perception in end-to-end autonomous driving. *J. Syst. Archit.* **2020**, *110*, 101766. [CrossRef]
7. Khan, M.A.; El Sayed, H.; Malik, S.; Zia, M.T.; Alkaabi, N.; Khan, J. A journey towards fully autonomous driving-fueled by a smart communication system. *Veh. Commun.* **2022**, *36*, 100476. [CrossRef]
8. Cheng, J.; Zhang, L.; Chen, Q.; Hu, X.; Cai, J. A review ofvisual SLAM methods for autonomous driving vehicles. *Eng. Appl. Artif. Intell.* **2022**, *114*, 104992. [CrossRef]
9. Su, Y.; Shan, S.; Chen, X.; Gao, W. Face Recognition: A Literature Survey. *IEEE Trans. Image Process.* **2009**, *18*, 1885–1896. [CrossRef]
10. Santra, B.; Mukherjee, D.P. A comprehensive survey on computer vision based approaches for automatic identification of products in retail store. *Image Vis. Comput.* **2019**, *86*, 45–63. [CrossRef]
11. Zhang, F.; Fu, L.S. Application of Computer Vision Technology in Agricultural Field. *Prog. Mechatron. Inf. Technol. PTS 1 2* **2014**, *462–463*, 72–76. [CrossRef]
12. Ardakani, A.A.; Kanafi, A.R.; Acharya, U.R.; Khadem, N.; Mohammadi, A. Application of deep learning technique to manage COVID-19 in routine clinical practice using CT images: Results of 10 convolutional neural networks. *Comput. Biol. Med.* **2020**, *121*, 103795. [CrossRef]
13. Minaee, S.; Kafieh, R.; Sonka, M.; Yazdani, S.; Jamalipour Soufi, G. Deep-COVID: Predicting COVID-19 from chest X-ray images using deep transfer learning. *Med. Image Anal.* **2020**, *65*, 101794. [CrossRef] [PubMed]
14. Zhang, W.; Tang, P.; Zhao, L. Remote sensing image scene classification using CNN-CapsNet. *Remote Sens.* **2019**, *11*, 494. [CrossRef]
15. Helber, P.; Bischke, B.; Dengel, A.; Borth, D. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2019**, *12*, 2217–2226. [CrossRef]
16. Cao, C.; Dragicevic, S.; Li, S. Land-Use Change Detection with Convolutional Neural Network Methods. *Environments* **2019**, *6*, 25. [CrossRef]
17. Naushad, R.; Kaur, T.; Ghaderpour, E. Deep Transfer Learning for Land Use and Land Cover Classification: A Comparative Study. *Sensors* **2021**, *21*, 8083. [CrossRef] [PubMed]
18. Rahnemoonfar, M.; Murphy, R.; Miquel, M.V.; Dobbs, D.; Adams, A. Flooded area detection from UAV images based on densely connected recurrent neural networks. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 1788–1791. [CrossRef]
19. 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, 1–14 December 2017; pp. 3746–3749. [CrossRef]
20. Lo, S.W.; Wu, J.H.; Lin, F.P.; Hsu, C.H. Visual sensing for urban flood monitoring. *Sensors* **2015**, *15*, 20006–20029. [CrossRef]
21. de Vitry, M.M.; Dicht, S.; Leitão, J.P. FloodX: Urban flash flood experiments monitored with conventional and alternative sensors. *Earth Syst. Sci. Data* **2017**, *9*, 657–666. [CrossRef]

22. Dhaya, R.; Kanthavel, R. Video Surveillance-Based Urban Flood Monitoring System Using a Convolutional Neural Network. *Intell. Autom. Soft Comput.* **2022**, *32*, 183–192. [CrossRef]
23. Jiang, J.; Liu, J.; Cheng, C.; Huang, J.; Xue, A. Automatic estimation of urban waterlogging depths from video images based on ubiquitous reference objects. *Remote Sens.* **2019**, *11*, 587. [CrossRef]
24. Li, Z.; Demir, I. U-net-based semantic classification for flood extent extraction using SAR imagery and GEE platform: A case study for 2019 central US flooding. *Sci. Total Environ.* **2023**, *869*, 161757. [CrossRef]
25. Pally, R.J.; Samadi, S. Application of image processing and convolutional neural networks for flood image classification and semantic segmentation. *Environ. Model. Softw.* **2022**, *148*, 105285. [CrossRef]
26. Ning, H.; Li, Z.L.; Hodgson, M.E.; Wang, C.Z. Prototyping a Social Media Flooding Photo Screening System Based on Deep Learning. *ISPRS Int. J. Geo-Inf.* **2020**, *9*, 104. [CrossRef]
27. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440. [CrossRef]
28. Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A Large-Scale Hierarchical Image Database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255. [CrossRef]
29. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
30. Ren, S.Q.; He, K.M.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 91–99. [CrossRef]
31. Sen, T.; Hasan, M.K.; Tran, M.; Yang, Y.; Hoque, M.E. Selective Search for Object Recognition. In Proceedings of the 13th IEEE International Conference on Automatic Face Gesture Recognition, (FG 2018), Xi'an, China, 15–19 May 2018; pp. 357–364. [CrossRef]
32. Shi, J.B.; Malik, J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 888–905. [CrossRef]
33. Khan, A.; Sohail, A.; Zahoora, U.; Qureshi, A.S. A survey of the recent architectures of deep convolutional neural networks. *Artif. Intell. Rev.* **2020**, *53*, 5455–5516. [CrossRef]
34. Andrew, H.; Mark, S.; Grace, C.; Liang-Chieh, C.; Bo, C.; Mingxing, T.; Weijun, W.; Yukun, Z.; Ruoming; Vijay, V. Searching for mobilenetv3. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 1314–1324.
35. Zhang, X.; Zhou, X.; Lin, M.; Sun, J. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6848–6856. [CrossRef]
36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 2016, pp. 770–778.
37. Wang, F.; Jiang, M.; Qian, C.; Yang, S.; Li, C.; Zhang, H.; Wang, X.; Tang, X. Residual Attention Network for Image Classification. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6450–6458. [CrossRef]
38. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; Volume 2019, pp. 10691–10700.
39. Ji, S.; Xu, W.; Yang, M.; Yu, K. 3D Convolutional Neural Networks for Human Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 221–231. [CrossRef] [PubMed]
40. Huang, G.; Liu, Z.; Pleiss, G.; van der Maaten Maaten, L.; Weinberger, K.Q. Convolutional Networks with Dense Connectivity. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 8704–8716. [CrossRef]
41. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *2017*, 5999–6009.
42. Shorten, C.; Khoshgoftaar, T.M. A survey on Image Data Augmentation for Deep Learning. *J. Big Data* **2019**, *6*, 1–48. [CrossRef]
43. Varol, G.; Laptev, I.; Schmid, C. Long-Term Temporal Convolutions for Action Recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 1510–1517. [CrossRef] [PubMed]