remote sensing

MDPI

Technical Note

Obtaining Urban Waterlogging Depths from Video Images Using Synthetic Image Data

Jingchao Jiang [1], Cheng-Zhi Qin [2,3], Juan Yu [4], Changxiu Cheng [5,*], Junzhi Liu [6] and Jingzhou Huang [1]

1 Smart City Research Center, School of Automation, Hangzhou Dianzi University, Hangzhou 310012, China; jiangjc@hdu.edu.cn (J.J.); huangjz@hdu.edu.cn (J.H.)
2 State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China; qincz@lreis.ac.cn
3 Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing 210023, China
4 College of Mathematics and Computer Science, Zhejiang Normal University, Jinhua 321004, China; yujuan@zjnu.edu.cn
5 State Key Laboratory of Earth Surface Processes and Resource Ecology, Beijing Normal University, Beijing 100875, China
6 Key Laboratory of Virtual Geographic Environment, Ministry of Education, Nanjing Normal University, Nanjing 210023, China; liujunzhi@njnu.edu.cn
* Correspondence: chengcx@bnu.edu.cn; Tel.: +86-10-5880-7241

Received: 30 January 2020; Accepted: 19 March 2020; Published: 22 March 2020

Abstract: Reference objects in video images can be used to indicate urban waterlogging depths. The detection of reference objects is the key step to obtain waterlogging depths from video images. Object detection models with convolutional neural networks (CNNs) have been utilized to detect reference objects. These models require a large number of labeled images as the training data to ensure the applicability at a city scale. However, it is hard to collect a sufficient number of urban flooding images containing valuable reference objects, and manually labeling images is time-consuming and expensive. To solve the problem, we present a method to synthesize image data as the training data. Firstly, original images containing reference objects and original images with water surfaces are collected from open data sources, and reference objects and water surfaces are cropped from these original images. Secondly, the reference objects and water surfaces are further enriched via data augmentation techniques to ensure the diversity. Finally, the enriched reference objects and water surfaces are combined to generate a synthetic image dataset with annotations. The synthetic image dataset is further used for training an object detection model with CNN. The waterlogging depths are calculated based on the reference objects detected by the trained model. A real video dataset and an artificial image dataset are used to evaluate the effectiveness of the proposed method. The results show that the detection model trained using the synthetic image dataset can effectively detect reference objects from images, and it can achieve acceptable accuracies of waterlogging depths based on the detected reference objects. The proposed method has the potential to monitor waterlogging depths at a city scale.

Keywords: urban flooding; waterlogging depth; video image; synthetic image data; reference object detection; convolutional neural network

1. Introduction

Urban flooding monitoring can provide key data for early warning and forecasting of urban flooding, as well as decision support for emergency response, so to mitigate loss of life and property during urban flooding events. Therefore, it is of great benefit to monitor urban flooding at city scales.

There are multiple data sources to monitor urban flooding, including water level sensors [1], remote sensing data [2,3], social-media/crowdsourcing data [4–6], and video surveillance data [7–10]. Among these data sources, social-media/crowdsourcing and video surveillance can record the process of urban flooding in image form and provide innovative means to monitor urban flooding.

Photos and video images from these two new data sources have been used for multiple applications. For example, Wang et al. [4] used landmarks as reference points to manually identify flood extent boundaries by comparing them to the Google Street View photos and satellite images. Wang et al. [5] used the crowdsourcing photos to automatically determine whether urban flooding occurs by employing a classification model with convolutional neural network (CNN). De Vitry et al. [10] used a deep convolutional neural network to detect floodwater in surveillance footage and proposed a novel qualitative flood index to monitor flood level trends. Diakakis et al. [11] used Google Street View to manually detect buildings that have been flooded. Schnebele et al. [12] used video data for a visual assessment of flood hazards. Photos sent by citizens were assessed by professionals to identify a clear water level around the time of the flood peak inundation [6].

Recently, obtaining waterlogging depths from video images has attracted research attention. In most of current research [13,14], waterlogging depths were manually estimated based on reference objects of video images, which is laborious and time-consuming for a city-scale application. In addition, the waterlogging depth data may not be immediately available or continuous during urban flooding events. How to automatically and efficiently obtain waterlogging depths from video images at a city scale should be explored. Automatic detection of reference objects from video images is the key.

There are two types of object detection models that can be used to automatically detect reference objects from video images, i.e., traditional object detection models and object detection models with CNNs. The traditional object detection models are based on handcrafted low-level features such as shift invariant feature transform (SIFT) [15], histogram of oriented gradients (HOG) [16], and so forth. The object detection models with CNNs are based on high-level abstract features automatically learning from CNNs, which are more complex than handcrafted features [17]. Object detection models with CNNs can achieve higher precision than traditional models. Furthermore, the ever-increasing powerful GPUs and lightweight deep learning models ensure models with CNNs can achieve object detection in real-time.

The edge detection operators can be used to detect reference-water interfaces [18,19]. In combination with pixel scale, waterlogging depths can be calculated based on reference-water interfaces. This method requires pre-/post-processing work and manually tunes the empirical parameters (e.g., thresholds). It shows some promise for obtaining urban waterlogging depths at a local scale (e.g., several monitoring points). However, it would be inappropriate and difficult for city-wide scale applications.

Compared with other methods, the object detection models with CNNs can detect objects directly from images at a city scale with little additional setting. Jiang et al. [9] proposed to utilize an object detection model with CNN to detect reference objects from video images and automatically calculate waterlogging depths based on detected reference objects. Object detection models with CNNs require a large number of labeled images as a training set, so to ensure the applicability and scalability of the models at a city scale. However, on the one hand, it may be not easy to collect a large number of urban flooding video images containing some valuable reference objects (e.g., traffic buckets) that are partly submerged in water. On the other hand, manual labeling of images is usually time-consuming and expensive. In fact, the lack of adequately labeled training image data is a common problem faced by CNN-based computer vision algorithms [20–22]. To address this deficiency, an alternative solution is to use some real or artificial images to generate synthetic image data as training image data for computer vision algorithms [23,24].

In recent years, synthetic image datasets have been used in object detection [25], optical flow estimation [26], pose estimation [27], text recognition [28], semantic segmentation [29], action recognition [30], and so forth. Prior work demonstrates the potential of synthetic image data as

training data to advance computer vision algorithms. Moreover, it is relatively easy to collect images containing reference objects and images with water surfaces from open data sources (e.g., image search engines and street views). With these types of images, it is promising to generate a sufficient number of synthetic images as training data in a short time. To the best of our knowledge, there is no synthetic image data of urban flooding scenes. It is attractive to create such synthetic image data to train object detection models for detecting reference objects from video images and further calculating waterlogging depths.

The objective of this paper is to propose a method of generating synthetic image data for training reference object detection models that can be used for city-wide scale applications, so to obtain waterlogging depths from video images using synthetic image data. The main contributions of this paper are three-fold:

(1) To the best of our knowledge, this is the first work to utilize synthetic images of urban flooding as training data for CNN-based reference object detection models that can be used at a city scale.
(2) Images containing reference objects and images about urban flooding are used to generate synthetic image data. Multiple data augmentation techniques are utilized to ensure the diversity and amount of synthetic image data.
(3) The effectiveness of synthetic image data on training the reference object detection model is evaluated.

The rest of this paper is organized as follows. Section 2 presents the details of the method and its further application in obtaining waterlogging depths. In Sections 3 and 4, the effectiveness of the proposed method is evaluated by a case study. The discussion is presented in Section 5, and the conclusions are presented in Section 6.

2. Method and Its Further Application

The flow chart of the proposed method and its further application in obtaining waterlogging depths is shown in Figure 1.

The method of generating a synthetic image dataset consists of four main steps in sequence: Firstly, images containing reference objects and images about urban flooding are collected. Secondly, reference objects and water surfaces are cropped form these images, respectively. Thirdly, data augmentations are made on images of reference objects and images of water surfaces, respectively. Lastly, images of reference objects are pasted on images of water surfaces to generate a synthetic image dataset.

The steps of using the synthetic image dataset to obtain waterlogging depths from video images are as follows: Firstly, an object detection model with CNN is trained using the synthetic image dataset. Secondly, reference objects are detected from video images by the trained detection model during the flooding periods. Lastly, the waterlogging depths are calculated using the height differences between the unsubmerged reference object detected during flooding periods and the whole reference object detected during non-flooding periods.

The proposed method and its further application in obtaining waterlogging depths are presented in detail in the following subsections.

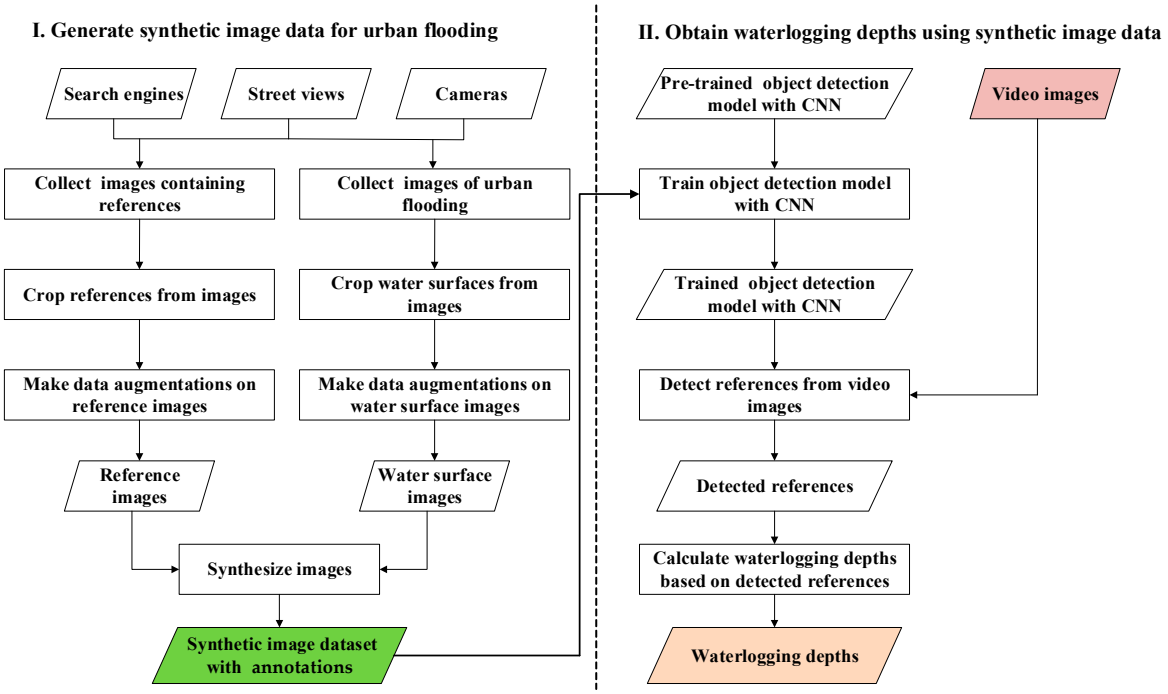

Figure 1. Flow chart of the proposed method and its further application in obtaining waterlogging depths.

2.1. Method of Generating Synthetic Image Dataset

2.1.1. Collection of Original Images, Cropping References, and Water Surfaces

Images containing reference objects (e.g., traffic buckets) can be collected from search engines and street views (Figure 2). A web crawler is used to automatically download images containing reference objects from search engines based on some keywords about reference objects. In addition, we took a few photos of reference objects with mobile phones.

Figure 2. Some images containing reference objects (e.g., traffic buckets) collected from search engines and street views.

In a similar way, images of urban flooding are collected from the search engines (Figure 3). The web crawler is used to automatically download images of urban flooding from search engines based on some keywords such as urban flooding and urban torrential rain. In addition, video images can also be collected from social software.

Figure 3. Some images of urban flooding collected from image search engines.

After the collection stage, all the collected images are manually checked, and all irrelevant or poor-quality images are finally discarded to ensure the quality of synthetic images.

Reference objects are manually matted from the original images containing reference objects to ensure the accuracy and the integrity of reference objects.

Water surfaces are manually cropped from the original images of urban flooding according to specifying crop rectangles.

2.1.2. Data Augmentations on Images of Reference Objects and Water Surfaces

Data augmentation is a widely used strategy to improve generalization capabilities of computer vision algorithms with CNNs and avoid over-fitting [31]. Color augmentations and geometry augmentations are usually used to generate images with sufficient diversity. In the proposed method, the color augmentations (including brightness alteration, contrast modification, color desaturation, and random Gaussian blurring) and the geometry augmentations (including flipping and resizing) are used on images of reference objects and water surfaces, respectively. In addition, truncation of reference object images is required for generating diversified samples, for the reason that reference objects are submerged in water during flooding periods and only parts of them are visible above water in most situations, whereas cropped reference objects from collected images are usually in complete shape.

2.1.3. Generating Synthetic Image Dataset with Annotations

One synthetic image is generated by pasting a reference object image on a water surface image. There are two ways of pasting images. The first way is to directly paste a reference object on a water surface, which can result in an unnatural boundary, as shown in Figure 4a. The second way is to blend a reference object on a water surface to smoothen the boundary by combining the pixels of the reference object with the pixels of the water surface, as shown in Figure 4b. The reference object is pasted in the center of the water surface image.

(**a**) (**b**)

Figure 4. Pasting a reference object (e.g., traffic bucket) on a water surface directly (**a**), and blending a reference object on a water surface in a natural way (**b**).

The annotation of each synthetic image is generated automatically. Taking the upper left corner of the water surface image as the origin point, the coordinates of the four corner points of the ground truth of the reference object are calculated. The information about reference object label and the coordinates of the four corner points are stored in a format (e.g., Extensible Markup Language, XML) that an object detection model with CNN can utilize.

2.2. Further Application of the Method: Using the Synthetic Image Dataset to Obtain Waterlogging Depths

2.2.1. Training an Object Detection Model

In recent years, many object detection models with CNNs have been proposed. Among them, the single-shot detector (SSD) [32] has been one of the state-of-the-art object detection models in terms of both accuracy and speed. Therefore, the SSD was used as the reference object detection model and trained using the synthetic image dataset with annotations. For more details regarding the SSD, please refer to reference [32].

2.2.2. Detection of Reference Objects from Video Images

The trained SSD is used to detect reference objects from video images during flooding periods and non-flooding periods. Video images are input into the SSD, and the category label and the four corner coordinates of the bounding box of each reference object are output by the SSD.

2.2.3. Calculation of Waterlogging Depths

The method to calculate waterlogging depths in reference [9] was used in this study. There are two main steps to calculate waterlogging depths. The first step is to calculate the heights of detected reference objects in pixel units during flooding periods and non-flooding periods. The height of the detected reference object in pixel units is calculated by the following equation:

$$h = y_{max} - y_{min} \tag{1}$$

where h is the height of the detected reference object in pixel units, y_{max} is the maximum value of the four corner coordinates of the bounding box in the y-axis in pixel units, and y_{min} is the minimum one.

The height of the submerged part of the reference object in pixel units during flooding periods is calculated in the following equation:

$$h_{dp} = h_p - h_{fp} \tag{2}$$

where h_{dp} is the height of the submerged part of the reference object in pixel units, h_p is the height of the entire reference object in pixel units during non-flooding periods, and h_{fp} is the height of the unsubmerged part of the reference object in pixel units.

The second step is to calculate waterlogging depths. Waterlogging depth is calculated in the following equation:

$$d = \frac{h_{dp}}{h_p} \times l \tag{3}$$

where d is the waterlogging depth, and l is the actual height of the entire reference object. l is obtained by actual measurement.

3. Materials

3.1. Video Image Dataset

In this study, a video dataset of a flooding event in a city in Hebei Province, China, was used to evaluate the effectiveness of the synthetic image dataset created by the proposed method. The video dataset recorded a pluvial event caused by a heavy rainfall on 21 July 2017. The traffic bucket that appeared in the video was used as the reference object. The height of the whole traffic bucket was

0.825 m. The width and height of the whole traffic bucket in pixel units during non-flooding periods were 19 and 30 pixels, respectively.

Considering potential ethical issues of using the video data, the sensitive/privacy information (e.g., faces, license plates, etc.) of video images was processed by the data provider before using the video data. The video was converted into 253 images in the joint photographic experts group (JPEG) format. Figure 5 shows sample partial images of the video. The traffic bucket of each image was manually labelled using the LabelImg software, and the ground truth bounding box of each traffic bucket was obtained. The heights of the traffic buckets detected manually were calculated according to Equation (1). These calculated heights were treated as the "observations" of the heights of the detected traffic buckets. The waterlogging depths were calculated according to Equations (2) and (3). These calculated waterlogging depths were treated as the "observations" of waterlogging depths due to lacking the observed values measured by water level sensors.

Figure 5. Sample partial images of the real video image dataset.

3.2. Computing Environment and SSD Model

The computing environment included a NVIDIA Corporation GM200GL (Tesla M40) and NVIDIA UNIX x86_64 Kernel Module 384.66 on a Linux system (i.e., Ubuntu 5.4.0-6ubuntu1~16.04.5) with an Intel(R) Xeon(R) CPU E5-2682 v4 at 2.50 GHz.

A specific SSD model (i.e., the SSD with MobileNet-V1) was trained using the synthetic image dataset. Then, the trained SSD model was used to detect traffic buckets from the real video image dataset.

4. Results

4.1. Synthetic Image Dataset

A synthetic image dataset was generated by the proposed method. The original images containing traffic buckets were collected from Baidu Image and photos taken by a mobile phone. Traffic buckets in many images were similar, so nine different traffic buckets were matted from these images. The original images of urban flooding were collected from Baidu Image and Google Picture. Sixteen different images of water surfaces were cropped from the original images. After utilizing the data augmentations in Section 2.1.2, 96 images of water surfaces and 54 images of traffic buckets were created. The size of the traffic buckets was 10 x 15 pixels, and the range of truncation was 0~1 pixel. A synthetic image was generated by pasting a traffic bucket on a water surface image in the blending way. A total of 10,368 synthetic images with annotations were generated. Figure 6 shows some sample images of the synthetic dataset.

Figure 6. Sample images of the synthetic dataset.

4.2. Accuracy and Efficiency

4.2.1. Accuracy of Heights of Detected Reference Objects

The parts of the images within a specific region of interest (RoI), rather than the entire images, were used to detect reference objects. The size of the RoI was 80 × 80 pixels. Traffic buckets in 244 images were detected by the trained SSD model. Figure 7 shows some examples of the detected results. Traffic buckets in nine other images were not detected by the SSD model, since vehicles or pedestrians in these images were mistakenly detected as reference objects.

Figure 7. Samples of the results of detecting traffic buckets from video images. The green box is the bounding box.

The heights of the detected reference objects were calculated based on the predicted bounding boxes of traffic buckets detected by the SSD model according to Equation (1). These calculated heights were treated as the "estimations" of the heights of the detected reference objects. Figure 8 shows that the observations and the estimations of the heights of the detected reference objects had acceptable agreement. The root-mean-square error (RMSE) was used to evaluate the accuracy of the heights of the detected reference objects. The RMSE value was 1.521 pixels.

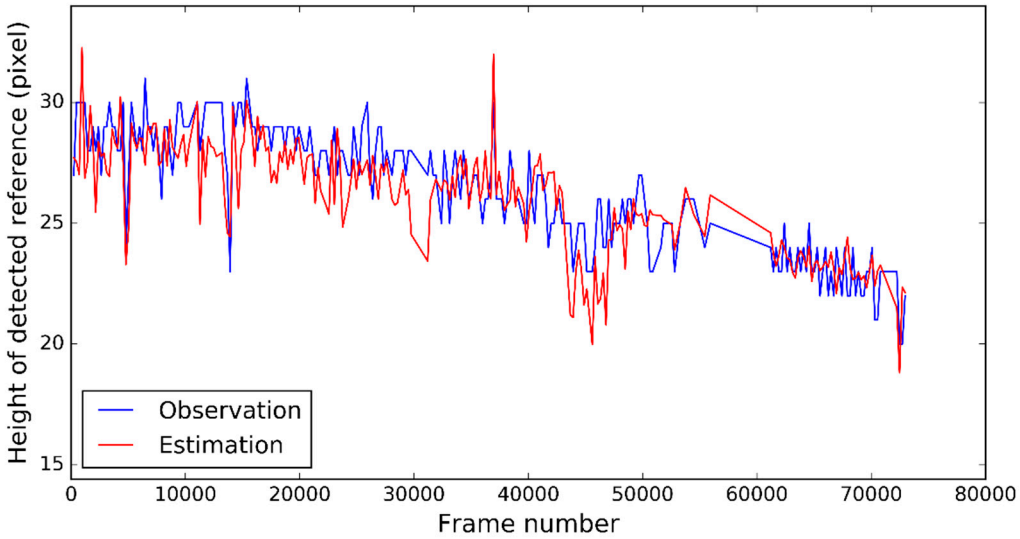

Figure 8. The heights of the reference objects detected manually (i.e., observations) and the heights of reference objects detected by the single-shot detector (SSD) (i.e., estimations).

4.2.2. Accuracy of Waterlogging Depths

The waterlogging depths were calculated based on the heights of detected reference objects according to Equations (2) and (3). The calculated waterlogging depths were treated as the "estimations" of the waterlogging depths. Figure 9 shows acceptable agreement between the observations and the estimations of waterlogging depths. The RMSE was used to evaluate the accuracy of waterlogging depths. The RMSE value was 0.041 m.

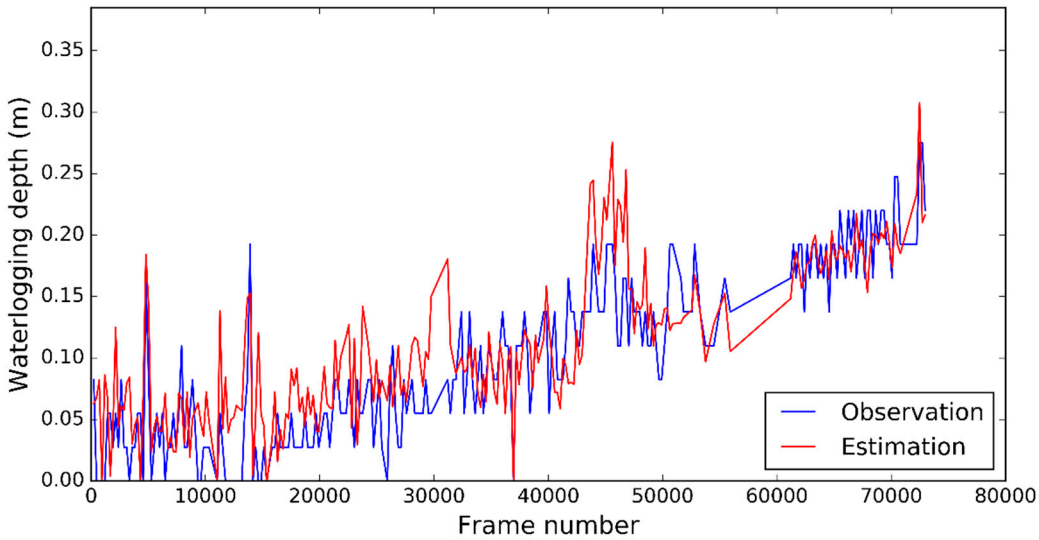

Figure 9. The waterlogging depths calculated based on the heights of the reference objects detected manually (i.e., observations) and the waterlogging depths calculated based on the heights of the reference objects detected by the SSD (i.e., estimations).

4.2.3. Efficiency of Labelling Reference Objects

It took about 27 s to generate the synthetic image set, including automatically generating 10,368 synthetic images and automatically labeling 10,368 traffic buckets.

As a reference, in this case study, it took about 2.5 h to manually label 253 traffic buckets from the real video images. This shows the high efficiency of the proposed method.

5. Discussion

5.1. Comparison with the Sensitivity Analysis Based on an Edge Detection Method

An image sensitivity analysis (i.e., reference-water interface, pixel scale) was used as the comparison method. The main steps of the sensitivity analysis were as follows. Firstly, the bounding box of the reference object was specified manually. Secondly, the image area of the reference object was cropped based on the bounding box. Thirdly, the cropped image was processed by using the Gaussian blur. In this step, the kernel size is a parameter that needs to be adjusted. Fourthly, the Canny operator was used to detect the reference-water interface. In this step, two thresholds needed to be adjusted. Fifthly, the vertical difference between the reference-water interface and the reference-road interface was calculated. Lastly, the vertical difference multiplied with the pixel scale was obtained as the waterlogging depth. It was found that the application of the sensitivity analysis was more tedious than that of the models with CNNs.

The real video images were used to evaluate the sensitivity analysis. The RMSE value of the vertical difference was 4.158 pixels, while the RMSE value of reference heights using the object detection model with CNN was 1.521 pixels. It was found that the accuracy of the sensitivity analysis was lower than that of the method with CNN.

5.2. Evaluation on an Artificial Image Dataset

Due to security and privacy issues, only one real video set was accessed and used in this study. To further demonstrate the effectiveness of this proposed method, an artificial image set (Figure 10) was used. The realistic images were created by using Adobe Photoshop and labelled by using LabelImg. The object detection model trained using the synthetic dataset was used to detect reference objects from these realistic images. The RMSE value of the heights of the detected reference objects was 1.384 pixels.

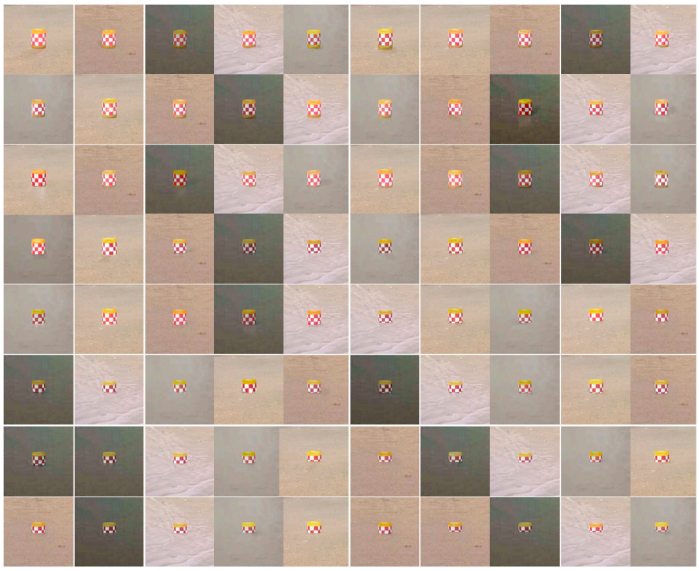

Figure 10. Images of the artificial image dataset.

5.3. Comparison Between the Synthetic Image Dataset and the Training Dataset of the Real Video Images

The K-fold cross validation was used to assess the performance of the SSD model trained using the real video images. The real video images with manual annotations in the case study were randomly divided into five subsets. For each subset, this subset was used as a testing dataset, while the remaining four subsets were used as a training dataset. For each testing dataset, the heights of the reference objects calculated based on the manual annotations were treated as the "observations", while the heights of

the reference objects detected by the SSD model trained using the corresponding training dataset were treated as the "estimations". The RMSE values of the heights of the detected reference objects in the five testing datasets were 0.872, 1.445, 1.349, 0.949, and 1.090 pixels, respectively. The average RMSE value of the heights of the detected reference objects was 1.141 pixels.

The RMSE values of the heights of the detected reference objects in the five testing datasets using the synthetic image dataset were 1.487, 1.433, 1.585, 1.221, and 1.818 pixels, respectively. The average RMSE value of the heights of the detected reference objects using the synthetic image dataset was 1.509 pixels. Therefore, the synthetic image dataset can obtain a comparable performance to that of the real video image dataset.

5.4. Effects of Data Augmentation Methods on the Effectiveness of Synthetic Datasets

The size of reference objects in synthetic images, truncation range, and the way in which reference objects are pasted on water surfaces are important parameters to generate synthetic image datasets by the proposed method. Their settings could affect the effectiveness of synthetic datasets. To assess their effects on the performance of the proposed method, we generated 16 synthetic image datasets under different sizes of reference objects in synthetic images, truncations ranges, and ways of pasting. It should be noted that the setting for other data augmentations of these synthetic datasets were the same. The SSD models were trained using these synthetic image datasets, respectively, then were used to detect reference objects from the real video images of the case study. The heights of the reference objects detected manually from the real video images were treated as the observations, and the heights of reference objects detected by the SSD models from the real video images were treated as the estimations. The RMSE values of the heights of the detected reference objects using different synthetic images datasets are shown in Table 1.

Table 1. The RMSE values of the detected reference object heights under different parameter settings.

Synthetic Datasets	Size (Pixels)	Truncation Range (Pixels)	Ways of Pasting	RMSE (Pixels)
S1	26 × 40	0	Direct Pasting	12.223
S2	26 × 40	0	Blending	9.847
S3	26 × 40	0~19	Direct Pasting	6.403
S4	26 × 40	0~19	Blending	4.427
S5	19 × 30	0	Direct Pasting	6.694
S6	19 × 30	0	Blending	5.849
S7	19 × 30	0~14	Direct Pasting	6.078
S8	19 × 30	0~14	Blending	3.441
S9	13 × 20	0	Direct Pasting	4.319
S10	13 × 20	0	Blending	1.733
S11	13 × 20	0~9	Direct Pasting	3.806
S12	13 × 20	0~9	Blending	2.794
S13	10 × 15	0	Direct Pasting	4.622
S14	10 × 15	0	Blending	2.286
S15	10 × 15	0~4	Direct Pasting	4.043
S16	10 × 15	0~4	Blending	2.775

It was found that the relatively small sizes of reference objects in synthetic images could achieve better accuracies than that of the big sizes, truncations of reference objects could improve the accuracies, and the errors of estimations in the blending way were smaller than that in the way of direct pasting.

5.5. Comparison with Other Related Research

Other related research was compared to expound the advantages of our proposed method. Wang et al. [5] employed a classification model with CNN (i.e., Clarifai) to automatically classify the crowdsourcing photos. Flooding was considered to happen if the flood tag was labeled in photos.

Although this type of qualitative information is useful, it is difficult to provide quantitative data for early warning and forecasting of urban flooding. De Vitry et al. [10] used a deep convolutional neural network (i.e., U-net) to detect floodwater in surveillance footage and proposed a novel qualitative flood index (SOFI) to monitor flood level trends. The training images in the study were collected from the Internet and manually labeled, and flood level trend was still a qualitative indicator. Jiang et al. [9] utilized an object detection model with CNN (i.e., SSD) to detect reference objects from video images, and they calculated waterlogging depths using the height differences between the detected reference object during non-flooding periods and the detected reference object during flooding periods. In this study, the training images needed to be manually labeled. Manually labeling images employed at a city scale is usually time consuming and expensive. Our proposed method used synthetic image data as the training data of CNN-based models. This method can generate a sufficient number of labeled images in a short time, which allows for urban flooding monitoring methods based on CNNs to be applied at a city scale. Therefore, compared with the other researchers, our method can provide real-time and wide-coverage quantitative monitoring data in an efficient and intelligent way.

5.6. Some Potential Issues

There are some sensitive/privacy information (e.g., human faces, license plates, etc.) in video images. The sensitive/privacy information should be processed in advance to avoid ethical issues. In addition, in order to better protect sensitive/privacy information, the proposed method can be prescribed for use on the government's dedicated intranet servers or government-regulated servers. Data security issues also need to be considered.

To ensure the reliability of the images collected from search engines, we manually checked these images and discarded irrelevant or poor-quality ones. Thereby, the images that were difficult to be rectified/ registered would not be used in our system. How to automatically ensure the reliability of images from search engines is a very important and interesting problem.

6. Conclusions

In this paper, we presented a method to generate synthetic image data for obtaining urban waterlogging depths from video images recording urban flooding. Firstly, original images containing reference objects and original images of urban flooding were collected from open data sources. Secondly, reference objects and water surfaces were cropped from original images. Thirdly, data augmentations were utilized to enrich images of reference objects and images of water surfaces. Lastly, synthetic images were generated via pasting images of reference objects on images of water surfaces. This method can generate abundant labeled image data with minimal manual effort for collecting images of urban flooding and labelling reference objects. A synthetic image dataset created by our method was used to train an object detection model (i.e., SSD). The SSD was employed to detect reference objects from a real video image dataset and an artificial image dataset, respectively. The results showed that acceptable accuracies of waterlogging depths were obtained using the synthetic images. It is found that the relatively small sizes of reference objects in synthetic images could achieve better accuracies, truncations of reference objects could improve the accuracies, and the blending way was more effective than the direct pasting way. It was also found that the synthetic image dataset can achieve comparable accuracy to the real video image dataset. The proposed method has the potential for city-scale applications, and it can provide continuous urban waterlogging depth information with wide coverage. The proposed method is easily portable and could be extended to synthesize image data for monitoring other types of disasters from video images.

Author Contributions: Conceptualization, J.J.; Methodology, J.J. and C.-Z.Q.; Software, J.J. and J.H.; Validation, J.J., J.L. and C.C.; Formal Analysis, C.-Z.Q.; Investigation, J.J.; Resources, J.L. and C.C.; Writing—Original Draft Preparation, J.J.; Writing—Review & Editing, C.-Z.Q., C.C. and J.Y.; Funding Acquisition, J.J., C.C. and J.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key Research and Development Plan of China (2019YFA0606901) and the National Natural Science Foundation of China (41601423; 41601413; 61702148).

Acknowledgments: The authors thank the Outstanding Innovation Team in Colleges and Universities in Jiangsu Province for processing data in this study.

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

References

1. Mousa, M.; Zhang, X.; Claudel, C. Flash flood detection in urban cities using ultrasonic and infrared sensors. *IEEE Sens. J.* **2016**, *16*, 7204–7216. [CrossRef]
2. Byun, Y.; Han, Y.; Chae, T. Image fusion-based change detection for flood extent extraction using bi-temporal very high-resolution satellite images. *Remote. Sens.* **2015**, *7*, 10347–10363. [CrossRef]
3. Li, L.; Xu, T.; Chen, Y. Improved urban flooding mapping from remote sensing images using generalized regression neural network-based super-resolution algorithm. *Remote. Sens.* **2016**, *8*, 625. [CrossRef]
4. Wang, Y.; Chen, A.S.; Fu, G.; Djordjević, S.; Zhang, C.; Savić, D.A. An integrated framework for high-resolution urban flood modelling considering multiple information sources and urban features. *Environ. Model. Softw.* **2018**, *107*, 85–95. [CrossRef]
5. Wang, R.Q.; Mao, H.; Wang, Y.; Rae, C.; Shaw, W. Hyper-resolution monitoring of urban flooding with social media and crowdsourcing data. *Comput. Geosci.* **2018**, *111*, 139–147. [CrossRef]
6. Le Coz, J.; Patalano, A.; Collins, D.; Guillén, N.F.; García, C.M.; Smart, G.M.; Bind, J.; Chiaverini, A.; Le Boursicaud, R.; Dramais, G.; et al. Crowdsourced data for flood hydrology: Feedback from recent citizen science projects in Argentina, France and New Zealand. *J. Hydrol.* **2016**, *541*, 766–777. [CrossRef]
7. Jiang, J.; Liu, J.; Qin, C.Z.; Wang, D. Extraction of urban waterlogging depth from video images using transfer learning. *Water* **2018**, *10*, 1485. [CrossRef]
8. Bhola, P.K.; Nair, B.B.; Leandro, J.; Rao, S.N.; Disse, M. Flood inundation forecasts using validation data generated with the assistance of computer vision. *J. Hydroinform.* **2019**, *21*, 240–256. [CrossRef]
9. 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]
10. De Vitry, M.M.; Kramer, S.; Wegner, J.D.; Leitão, J.P. Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network. *Hydrol. Earth Syst. Sci.* **2019**, *23*, 4621–4634. [CrossRef]
11. Diakakis, M.; Deligiannakis, G.; Pallikarakis, A.; Skordoulis, M. Identifying elements that affect the probability of buildings to suffer flooding in urban areas using Google Street View. A case study from Athens metropolitan area in Greece. *Int. J. Disaster Risk Reduct.* **2017**, *22*, 1–9. [CrossRef]
12. Schnebele, E.; Cervone, G.; Waters, N. Road assessment after flood events using non-authoritative data. *Nat. Hazards Earth Syst. Sci.* **2014**, *14*, 1007–1015. [CrossRef]
13. Fohringer, J.; Dransch, D.; Kreibich, H.; Schröter, K. Social media as an information source for rapid flood inundation mapping. *Nat. Hazards Earth Syst. Sci.* **2015**, *15*, 2725–2738. [CrossRef]
14. Liu, L.; Liu, Y.; Wang, X.; Yu, D.; Liu, K.; Huang, H.; Hu, G. Developing an effective 2-d urban flood inundation model for city emergency management based on cellular automata. *Nat. Hazards Earth Syst. Sci.* **2015**, *15*, 381–391. [CrossRef]
15. Lowe, D.G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **2004**, *60*, 91–110. [CrossRef]
16. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–25 June 2005; pp. 886–893.
17. Zhao, Z.-Q.; Zheng, P.; Xu, S.; Wu, X. Object detection with deep learning: A review. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 3212–3232. [CrossRef]
18. Gilmore, T.E.; Birgand, F.; Chapman, K.W. Source and magnitude of error in an inexpensive image-based water level measurement system. *J. Hydrol.* **2013**, *496*, 178–186. [CrossRef]
19. Nguyen, L.S.; Schaeli, B.; Sage, D.; Kayal, S.; Rossi, L. Vision-based system for the control and measurement of wastewater flow rate in sewer systems. *Water Sci. Technol.* **2009**, *60*, 2281–2289. [CrossRef]
20. Peng, X.; Sun, B.; Ali, K.; Saenko, K. Learning deep object detectors from 3d models. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1278–1286.

21. Wang, K.; Gou, C.; Zheng, N.; Rehg, J.M.; Wang, F.Y. Parallel vision for perception and understanding of complex scenes: Methods, framework, and perspectives. *Artif. Intell. Rev.* **2017**, *48*, 299–329. [CrossRef]
22. Zhan, F.; Lu, S.; Xue, C. Verisimilar image synthesis for accurate detection and recognition of texts in scenes. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018; pp. 257–273.
23. Alhaija, H.A.; Mustikovela, S.K.; Mescheder, L.; Geiger, A.; Rother, C. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *Int. J. Comput. Vis.* **2018**, *126*, 961–972. [CrossRef]
24. Gaidon, A.; Lopez, A.; Perronnin, F. The reasonable effectiveness of synthetic visual data. *Int. J. Comput. Vis.* **2018**, *126*, 899–901. [CrossRef]
25. Pepik, B.; Benenson, R.; Ritschel, T.; Schiele, B. What is holding back convnets for detection? In Proceedings of the German Conference on Pattern Recognition, Aachen, Germany, 7–10 October 2015; pp. 517–528.
26. Mayer, N.; Ilg, E.; Fischer, P.; Hazirbas, C.; Cremers, D.; Dosovitskiy, A.; Brox, T. What makes good synthetic training data for learning disparity and optical flow estimation? *Int. J. Comput. Vis.* **2018**, *126*, 942–960. [CrossRef]
27. Chen, W.; Wang, H.; Li, Y.; Su, H.; Wang, Z.; Tu, C.; Lischinski, D.; Cohen-Or, D.; Chen, B. Synthesizing training images for boosting human 3D pose estimation. In Proceedings of the International Conference on 3D Vision, Stanford, CA, USA, 25–28 October 2016; pp. 479–488.
28. Jaderberg, M.; Simonyan, K.; Vedaldi, A.; Zisserman, A. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vis.* **2016**, *116*, 1–20. [CrossRef]
29. Richter, S.R.; Vineet, V.; Roth, S.; Koltun, V. Playing for data: Ground truth from computer games. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 102–118.
30. Rahmani, H.; Mian, A. Learning a non-linear knowledge transfer model for cross-view action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 2458–2466.
31. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. In Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1106–1114.
32. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multibox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 8–16 October 2016; pp. 21–37.