Estimating Rainfall Intensity Using an Image-Based Deep Learning Model

Yin, Hang ; Zheng, Feifei; Duan, Huan-Feng; Savić, Dragan; Kapelan, Zoran

DOI
10.1016/j.eng.2021.11.021

Publication date
2022

Document Version
Final published version

Published in
Engineering

Citation (APA)
Yin, H., Zheng, F., Duan, H.-F., Savić, D., & Kapelan, Z. (2022). Estimating Rainfall Intensity Using an Image-Based Deep Learning Model. *Engineering*, *21*, 162-174. https://doi.org/10.1016/j.eng.2021.11.021

Contents lists available at ScienceDirect

Engineering

journal homepage: www.elsevier.com/locate/eng

Research
Hydraulic Engineering—Article

Estimating Rainfall Intensity Using an Image-Based Deep Learning Model

Hang Yin [a], Feifei Zheng [a,*], Huan-Feng Duan [b], Dragan Savic [c,d,e], Zoran Kapelan [d,f]

[a] College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310003, China
[b] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong 999077, China
[c] KWR Water Research Institute, Nieuwegein 3430 BB, The Netherlands
[d] Centre for Water Systems, University of Exeter, Exeter EX4 4QF, UK
[e] Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Malaysia
[f] Department of Water Management, Delft University of Technology, Delft 2600 GA, The Netherlands

ARTICLE INFO

Article history:
Received 9 July 2021
Revised 27 September 2021
Accepted 15 November 2021
Available online 27 January 2022

Keywords:
Urban flooding
Rainfall images
Deep learning model
Convolutional neural networks (CNNs)
Rainfall intensity

ABSTRACT

Urban flooding is a major issue worldwide, causing huge economic losses and serious threats to public safety. One promising way to mitigate its impacts is to develop a real-time flood risk management system; however, building such a system is often challenging due to the lack of high spatiotemporal rainfall data. While some approaches (i.e., ground rainfall stations or radar and satellite techniques) are available to measure and/or predict rainfall intensity, it is difficult to obtain accurate rainfall data with a desirable spatiotemporal resolution using these methods. This paper proposes an image-based deep learning model to estimate urban rainfall intensity with high spatial and temporal resolution. More specifically, a convolutional neural network (CNN) model called the image-based rainfall CNN (irCNN) model is developed using rainfall images collected from existing dense sensors (i.e., smart phones or transportation cameras) and their corresponding measured rainfall intensity values. The trained irCNN model is subsequently employed to efficiently estimate rainfall intensity based on the sensors' rainfall images. Synthetic rainfall data and real rainfall images are respectively utilized to explore the irCNN's accuracy in theoretically and practically simulating rainfall intensity. The results show that the irCNN model provides rainfall estimates with a mean absolute percentage error ranging between 13.5% and 21.9%, which exceeds the performance of other state-of-the-art modeling techniques in the literature. More importantly, the main feature of the proposed irCNN is its low cost in efficiently acquiring high spatiotemporal urban rainfall data. The irCNN model provides a promising alternative for estimating urban rainfall intensity, which can greatly facilitate the development of urban flood risk management in a real-time manner.

1. Introduction

Occurrences of urban floods can be attributed to many factors, including intensified rainfall due to a changing climate [1], increased surface runoff due to urbanization [2], and complex interactions between urban runoff and high water levels downstream (i.e., compound flooding events) [3]. Rainfall extremes often lead to urban floods—especially flash floods, for which rainfall intensity is the dominant factor [4]. It is notable that flash flooding can often result in more serious consequences compared with other flood types (e.g., river flooding), including a high number of casualties [5]. Such consequences mainly occur because the rainfall process associated with urban flash flooding is characterized by its sudden large intensity, which often leads to inadequate preparation of flood defense resources and delayed evacuation [6].

To mitigate the impacts of urban floods, a number of different solutions have been proposed over the past few decades [7–9]. One way is to develop a real-time urban flood warning system to enable an accurate inundation prediction, which would allow flooding-defense resources and evacuation to be operated in a timely manner [10]. However, a significant challenge associated with real-time urban flood warning systems is the lack of urban rainfall data with high spatiotemporal resolution [11,12]. This lack of data is because the urban rainfall process is often complex, as it is not only affected by large-scale land–ocean interactions, but also by local meteorological evolution [13]. As a result, urban rainfall events often exhibit complicated temporal and spatial distribution

[*] Corresponding author.
E-mail address: feifeizheng@zju.edu.cn (F. Zheng).

https://doi.org/10.1016/j.eng.2021.11.021

properties. For example, Berg et al. [14] and Wasko and Sharma [15] have stated that the temporal distribution of many observed rainfall events has become steeper in the changing climate, and the intensity can vary significantly in a short time period (e.g., 10 min). In terms of rainfall spatiality, it has been reported that the relative difference in rainfall intensity between two locations with a 3–5 km spatial distance can be up to 30%–50% [16–18].

The spatiotemporal properties of urban rainfall extremes can significantly affect the distribution characteristics of urban flooding, which include the inundation extent, water depths, and flood timing in different urban regions [19,20]. Therefore, to account for the spatiotemporal characteristics of rainfall events, it is important and necessary to develop a real-time urban flood warning system [21]. Such a system uses real-time rainfall data (i.e., 1 min resolution) with a high spatial resolution across the entire city (i.e., 100 m × 100 m), with which inundation predictions can be temporally and spatially accurate. These accurate inundation predictions can be subsequently employed to enable the effective operation of flood defense resources and the development of an efficient evacuation strategy.

More specifically, a real-time urban flood warning system consists of real-time rainfall data and an efficient hydrologic–hydraulic modeling module. The latter is less of a challenge, due to rapid developments in computing techniques in recent years [22]. A number of different methods are available to acquire or predict urban rainfall data. These methods can be classified into two types: model-based and equipment-based approaches. Model-based methods, such as the weather research and forecasting model (WRF) [23] or global climate models (GCM) [24], are typically unable to provide accurate rainfall estimates with high spatiotemporal resolution at an urban scale [25]. Equipment-based approaches include ground rainfall stations [26], weather radar [27], and satellite remote sensing [28]. Ground rainfall stations can measure rainfall data accurately, but often with low spatial resolution due to the limited number of stations in an urban area [29]. Weather radar can predict rain intensity with a high temporal resolution based on the scattering effect of electromagnetic waves [30]. However, the prediction accuracy of the weather radar approach cannot be guaranteed due to a number of influencing factors, such as uneven vertical distribution of rainfall, anomalous propagation of electromagnetic waves, and high buildings [27]. More importantly, since the number of ground-based radar stations is often low in many countries, the spatial coverage afforded by this approach is often limited. In contrast, the satellite remote sensing approach can provide rainfall prediction at a large spatial coverage, but its spatiotemporal resolution is often insufficient at the urban scale [31].

In recent years, crowdsourcing methods have been considered as an alternative way to collect rainfall data, including the uses of smart wipers in moving cars (e.g., Tesla cars) [32] or intelligent umbrellas with acoustic sensors [33]. Recently, a new approach was proposed by Jiang et al. [34] to measure rainfall intensity based on videos acquired by ordinary surveillance cameras. More specifically, the researchers developed a convex optimization algorithm to effectively decompose rainy images, followed by rainfall intensity estimates through geometrical optics and photographic analyses [34]. While these crowdsourcing methods are interesting, their wide application in acquiring rainfall data with high spatiotemporal resolution is difficult due to the associated high implementation complexities [32].

This study proposes a novel approach to measure urban rainfall data with a high spatiotemporal resolution using an image-based deep learning model. The proposed approach is motivated by the facts that: ① Images of rainfall events are widely available in cities, as they can be acquired from transportation cameras, security cameras, and smart phones at very low cost; and ② rainfall images that are highly temporally and spatially distributed across the entire city can be obtained by using existing sensors (cameras) or via citizen science (smart phones). In this work, a deep learning approach called a convolutional neural network (CNN) is adapted to predict rainfall intensity based on images collected from urban sensors. In recent years, deep learning methods have been wildly used in the field of environmental remote sensing [35] and earth systems science [36], demonstrating their great potential for solving traditional challenges in these fields. Among these deep learning methods, CNNs have been increasingly used in the hydro-meteorology field, for applications that include increasing the prediction accuracy of El Niño occurrence [37], predicting cyanobacteria concentrations in river water [38], extracting the velocity and pressure field from flow field images [39], and accelerating urban flood model computations [40]. However, to the best of our knowledge, this is the first work in which CNNs have been adapted to model rainfall with high spatiotemporal resolution based on urban sensors.

The most important feature of the proposed method is its extremely low cost in acquiring highly spatiotemporal urban rainfall data, which makes the development of a real-time flooding warning system possible. In addition, these rainfall data can be used to understand how climate change and urbanization affect the local hydrological cycle on an urban scale. It is anticipated that the proposed rainfall-estimating method will be promising for mitigating the impacts of urban floods—especially flash floods—as the assimilation and integration of various types of urban sense data are a growing trend in recent years toward urban "digital twins."

The remainder of this paper is organized as follows. Section 2 introduces the methodology of the proposed method and provides an outline of the proposed image-based rainfall CNN (irCNN) model architecture. Section 3 introduces the data for model development, and Section 4 discusses model training and validation. The last section presents the results and discussions.

2. Methodology

2.1. The methodological framework

Fig. 1 illustrates the overall concept of the proposed method, which includes the collection of rainfall images and model development and application. In a rainfall event, a large number of images are first collected from existing sensors—mainly public cameras that have been widely installed in cities (Fig. 1). Subsequently, the proposed irCNN model is employed to predict the rainfall intensity based on these images. Finally, the rainfall intensity for each analyzed location (i.e., each location that

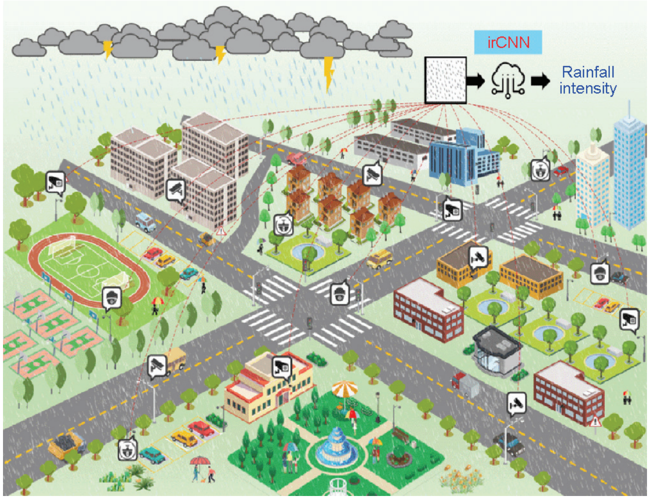

Fig. 1. The concept of the proposed method.

provides rainfall images) is obtained, resulting in rainfall data with a high spatiotemporal resolution.

The development of an image-based rainfall model is challenging due to the following issues: ① Rainfall images from different urban locations have different backgrounds; and ② the background of a single location can vary due to the weather and changes in the environment status (e.g., traffic). Fortunately, the CNN model has exhibited great ability in image recognition, as demonstrated in the domain of artificial intelligence [41,42]. Therefore, a CNN model framework is adopted in this study.

The overall procedure of the proposed method includes the model setup, data acquisition for model development, and model training and validation. Within the model setup stage, the irCNN framework is proposed, in which a regression layer is added to the existing CNN architecture to enable the generation of continuous values in the results. Given that the number of CNN parameters is large, an open-source ImageNet dataset is used to pretrain the irCNN before its use to estimate rainfall intensity. Subsequently, rainfall data is collected for irCNN model development and conditioned on the pretrained framework, with data sources including synthetic rainfall images, images from smart phones, and images from *in situ* cameras. These data are then used to further train the irCNN. Finally, the model's accuracy in simulating rainfall intensity is validated.

2.2. The setup of the imaged-based rainfall CNN model

2.2.1. The CNN model

A CNN is a typical deep learning method that was initially developed for document and image recognition [43]. The CNN model is a representation learning-based method, characterized by using multiple levels of representations (i.e., parameters) to represent different feature levels. More specifically, the CNN model can be fed with raw data and automatically discovers the representations needed for detection or classification.

Fig. 2 shows the typical architecture of a CNN model, which often includes an input layer, convolutional layers, subsampling layers, full connection layers, and an output layer. To explain the process used by a CNN, the following example is given, which applies a CNN to identify the number "8." As shown in Fig. 2, the input plane receives an image with a number (represented by a pixel matrix) that is approximately size normalized [41]. Next, multiple feature maps with different weight vectors are generated within the convolution layer using a set of different convolutional kernels (often using a 3 × 3 matrix). Subsequently, the subsampling layer is used to carry out a local averaging or maxima in order to reduce the resolution of the feature map and the sensitivity of the output to the shift and distortion of the original input. The convolution and the subsampling processes must be performed many times to identify the features of the input image. Finally, full collection layers are employed to generate the output (a probability vector for the number from 0 to 9) based on the feature maps. It should be noted that the number of featured maps at each convolutional layer and subsampling layer need to be prespecified based on rules. The details of a CNN architecture can be found in Ref. [43].

Due to the rapid development of computing power over the past few decades, a number of CNN model variants have been proposed in the research area of computer science and engineering, including AlexNet [44], visual geometry group networks (VGGs) [45], and residual networks (ResNets) [42]. These models have been demonstrated to be effective and efficient in classifying images and detecting objects within complex images [46,47].

2.2.2. The irCNN model

Based on the overall framework of the typical CNN architecture, the present study proposes an irCNN model with the aim of estimating rainfall intensity based on rainfall images. From an intuitive perspective, rainfall intensity can be represented by the density and size of raindrops in an image. In other words, a rainfall image with a relatively large raindrop density and size can be generally associated with greater rainfall intensity, and vice versa. This relationship can be mathematically expressed as follows:

$$I = f(Z(d, s)) \tag{1}$$

where I is the rainfall intensity ($\mathrm{mm \cdot h^{-1}}$); Z is the rain image; d and s are the density and size of raindrops, respectively; and f represents the underlying nonlinear relationship between the rain image and rainfall intensity, which must be derived using a CNN model.

In the literature, a number of different CNN types have been developed, which differ in terms of their model architectures, such as the number of layers, the size of convolutional kernels, the manner of enabling subsampling, and so on; further details are provided in the work of He et al. [42]. In this study, the irCNN model is based on the ResNet34 model type (where 34 indicates a total of 34 layers in the model), which has been widely used in computer science and engineering [42]. More importantly, the ResNet34 model has been demonstrated to perform better than other alternatives, such as AlexNet (developed in 2012), VGG 16 (developed in 2014), and Graph NN, in many applications [48,49]. Fig. 3 outlines the irCNN model architecture with its total of 38 layers (including additional input and output layers). As shown in this figure, the ResNet34 model process starts with an input image with a photo ($L = 1$, where L is the layer number in the irCNN model). In the second layer ($L = 2$), 64 convolutional kernels (each of which is a 7 × 7 matrix) are used to generate 64 different feature maps (planes in Fig. 2) for the original input image. However, it should be noted that the convolutional kernel is applied to the pixel matrix of the input image by moving every two columns (convolutional stride $s = 2$) in a subsampling process to reduce the resolution of the feature map. Therefore, the convolution and subsampling are jointly performed in the second layer ($L = 2$), which is also the case for L = 10, 18, and 30, as represented by the green blocks in Fig. 3.

In the irCNN model architecture (Fig. 3), a total of 29 convolutional layers (excluding the layers with the subsampling process) are used (yellow blocks), with an identical convolutional kernel

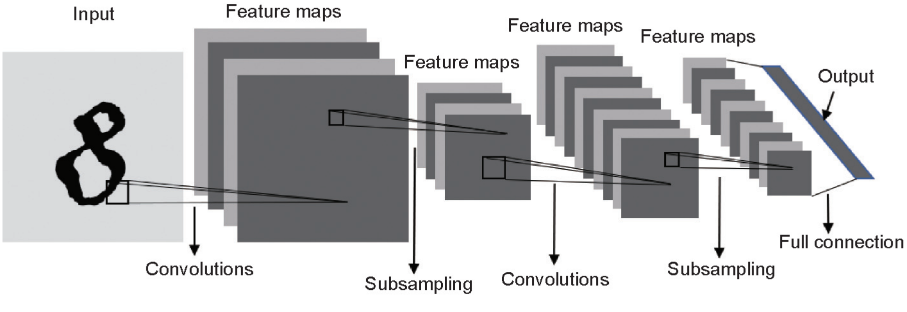

Fig. 2. Architecture of a typical CNN used for digit recognition, where each plane is a feature map.

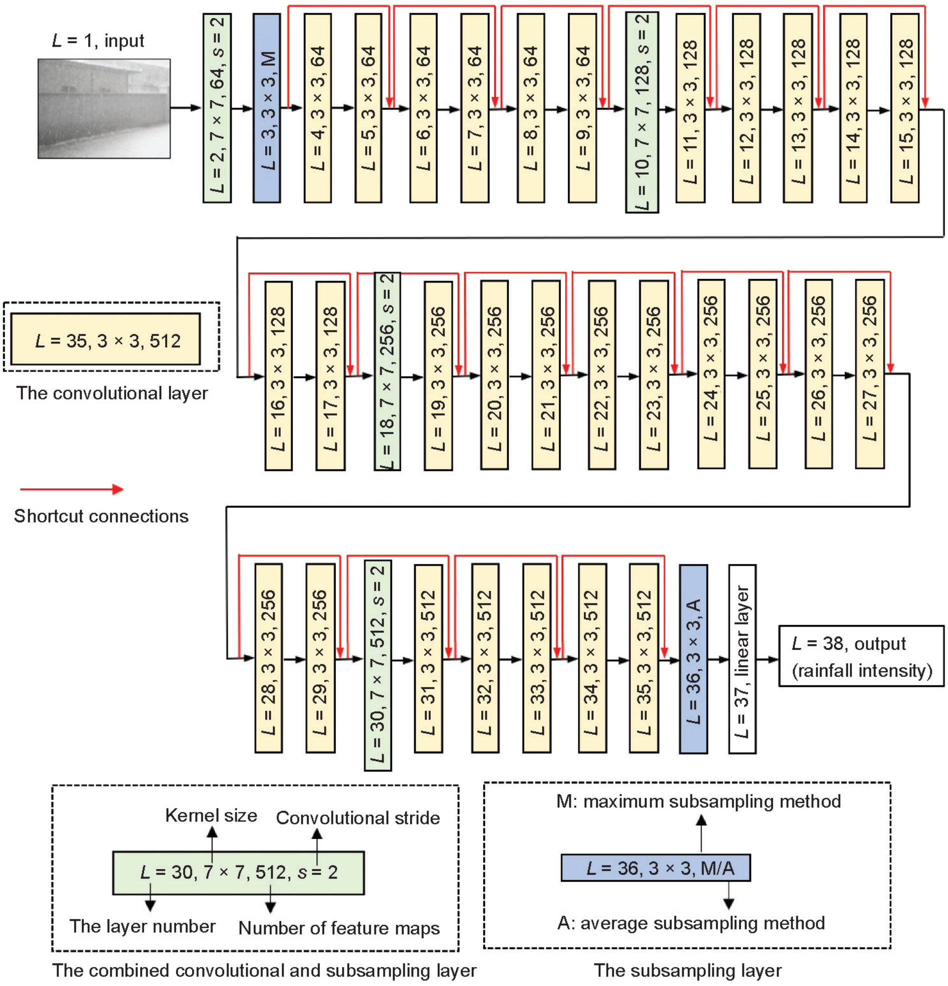

Fig. 3. The architecture of the proposed irCNN model.

size (a 3 × 3 matrix) but an increasing number of feature maps (ranging from 64 to 512). Two special subsampling layers (L = 3 and 36) are used in the irCNN model (blue blocks in Fig. 3), where L = 3 and 36 respectively identify the maximum (maximum subsampling method) and average (average subsampling method) value of each 3 × 3 matrix of the feature maps. It is notable that the values of each convolutional kernel used in Fig. 3 must be calibrated, resulting in a total of 6.3×10^7 parameters for calibration. This particular architecture of the ResNet34 model was suggested by He et al. [42] based on a comprehensive simulation analysis.

Within the irCNN model architecture, a deep residual learning approach is used, following the work of He et al. [42], in order to address the degradation problem (i.e., premature convergence) that often exists in CNN model applications. More specifically, shortcut connections (red arrow lines in Fig. 3) are involved in the irCNN model architecture, where the shortcut connections skip two layers, as shown in Fig. 3. These shortcut connections simply perform identity mapping, and their outputs are added to the outputs of the stacked layers (Fig. 3).

Typical CNN models are mainly used for object classification; hence, a regression layer (L = 37 in Fig. 3) is added to the proposed irCNN model in order to estimate rainfall intensity. This adaption can be expressed as follows:

$$\hat{I} = \boldsymbol{W}^{\mathrm{T}}\boldsymbol{X} + b \tag{2}$$

where $\hat{I}$ represents the predicted rainfall intensity; $\boldsymbol{W}$ represents the parameters of the linear regression layer, which will be automatically determined within the training process of the irCNN model; $\boldsymbol{X}$ represents the output (a vector) of the previous layer (L = 36); and b represents the bias term.

Since the number of irCNN model parameters to be calibrated is large, sufficient model training requires a huge number of rainfall images, which was not available in the present study. To solve this problem, an open-source dataset called ImageNet, consisting of 1000 classes with 1.28 million images, was used to pretrain the irCNN model to classify objectives (e.g., different animals) in these images [50]. In other words, the irCNN model is first trained on images unrelated to rainfall, as they are largely available; hence, approximate values of irCNN parameters can be obtained after the pretraining process [50]. Once this is done, follow-up training based on rainfall image data (Section 3) determines the final CNN parameter values. The above approach is often used in the CNN domain [42] because—despite the images containing different types of objects—all these objects share certain common features that are relevant for their detection and classification.

3. Rainfall data acquisition for irCNN model development

3.1. Synthetic rain images

A total of 4000 different publicly available images [51] without raindrops are used to generate the synthetic rainfall images. These public images are considered as the background layer; the image-processing software Photoshop CC2017 [52] is then used to add the raindrop layer (i.e., noise layer) to these images. Within the raindrop layer, a range of different raindrop densities, sizes, and angles (i.e., wind impact) can be considered to generate a sufficient diversity of rainfall events. After a preliminary analysis and given that the rainfall intensity is dominated by the raindrop density and size (Eq. (1)), we assume the following mapping relationship between the raindrop density, size, and intensity:

$$\mathrm{SI} = 100d \times \frac{1}{16}s^2 \tag{3}$$

where SI is the synthetic rainfall intensity, which is dimensionless, and d represents the raindrop density on the background layer, which is defined as the ratio of the pixels of the noise points relative to the total pixels. In Photoshop CC2017 [52], d ranges from 0.1% to 100%; hence, $100d$ ranges from 0.1 to 100. In this study, d is restricted to within 10%–19% with a resolution of 1%, resulting in a total of 10 different SI values. The restriction of d to within 10%–19% is because the noise density of such a range is similar to that of the real rainfall images through visualization; however, other different d values can also be easily applied to the proposed irCNN model.

In Eq. (3), s represents the raindrop size, which is defined as the ratio of the area of the raindrop layer to the area of the background layer [52]. In this study, three different values of s—namely, 350%, 400%, and 450%—are used for raindrop sizes; of these, s = 400% is considered to be the default raindrop size, as this size was judged visually to be the most appropriate in comparison with a real raindrop size [52].

We use the following strategy to change the sizes of the raindrops in the rain images: First, the raindrop layer is superposed onto the background layer of an image with a particular background using s = 400%. The raindrop size s can then be increased or decreased with a change rate of k by enhancing or shrinking the area of the raindrop layer, where the area change rate is k^2. This change leads to a corresponding decrease or increase in the value of d with a rate of k^2 as the number of raindrops in the background layer is changed as a result of the area variation of the raindrop layer. However, the SI value is not varied within the process indicated by Eq. (3). For example, when s is increased from 400% to 450%, the area of the raindrop layer is increased by $\left(\frac{45}{40}\right)^2 = \frac{81}{64}$, and the raindrop density on the background layer is reduced by $\frac{64}{81}$; thus, the SI value stays the same (Eq. (3)).

Eq. (3) is used to generate rainfall intensities for synthetic rainfall images, which are used accordingly as the data labels for irCNN model developments. Fig. 4 shows three synthetic rainfall images that are produced for the same background layer but have different raindrop densities. For a given raindrop density, its distribution and relative size follow a Gaussian distribution, as stated in the tutorial for Photoshop CC2017 [52]. While Eq. (3) is assumed in this study to develop a mapping between raindrop density, size, and rainfall intensity, other mapping equations can easily be applied to generate synthetic rainfall images.

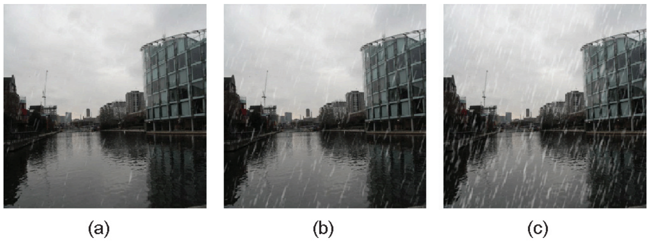

Fig. 4. Examples of synthetic rain images with the effect of increasing SI values: (a) SI = 10; (b) SI = 13; and (c) SI = 18.

Two different synthetic datasets are generated using the method described above, with background images taken from Ref. [51]. The details of the two synthetic datasets are presented below.

Synthetic dataset 1 (SD1): This dataset is used to investigate how an increase of background diversity affects the model performance. In SD1, the validation sub-dataset consists of 1000 rainfall images with different backgrounds. These are produced by using 100 different background images, with each image being superposed by a noise layer (i.e., a rain layer) with ten different SI values (from 10 to 19, with a resolution of 1). For the training sub-dataset, the number of background images (N) gradually increases from 100 to 1200 with a resolution of 100; the background images are randomly selected from the total background dataset. Here, it should be noted that each background image can only be selected once for a particular N, and the images used in the validation data are not used in the training sub-dataset. Subsequently, for each background image, ten different SI values are used to generate the rainfall images, resulting in 12 training sub-datasets in SD1. For example, if N = 500, then a total of 5000 rainfall images are produced, all with different backgrounds. Additional details are provided in Table 1.

Synthetic dataset 2 (SD2): This dataset is used to investigate how the increase of diversity in the rainfall intensity influences the predictive performance. The validation sub-dataset of SD2 is identical to that used in SD1. For the training sub-dataset, the number of background images is fixed; a range of different rainfall intensity combinations (the set of $\boldsymbol{C}$) of varying size are selected from the ten SI values to generate rainfall images. More specifically, in this study, the size of $\boldsymbol{C}$ increases from 2 (a combination of two SI values) to 10 (the combination of all available SI values) with a resolution of 1. For each SI value in $\boldsymbol{C}$, the fixed 800 background images are jointly used to produce synthetic images using the method described above. Additional details of SD2 are given in Table 2.

3.2. Real rainfall images from smart phones

To further validate the performance of the irCNN model, real rainfall images were collected using smart phones during rainfall events on the campus of Zhejiang University, China, as shown in Fig. 5. A tipping-bucket rain gauge with 1 min time resolution and 0.1 mm rainfall intensity precision was installed on campus at the location shown in Fig. 5. During rainfall events, portable smart phones were used to capture rainfall images at different locations, as shown in Fig. 5.

It should be noted that the rainfall gauge recorded the accumulated rainfall with a resolution of every 1 min, but the exposure time of the photos was very short (around 1/200 s); this resulted in a potential mismatch between the rainfall recorded at a gauge and the true intensity captured by a photo on a smart phone. To address this issue, we use a linear interpolation method to estimate the rainfall intensity. A linear interpolation is used in this work due to its great simplicity; however, future research should develop and use more advanced rainfall downscaling methods to further improve the predictive performance of the irCNN model. As illustrated in Fig. 6, the rainfall intensity at each intermediate time (I_L at time t_L and I_R at time t_R) of the recording time interval (from T_0 to T_0 + Δt, where Δt = 1 min in this study) is assumed to be the intensity measured by the accumulated rainfall depth at the end of the time interval (T_0 + Δt, T_0 + 2Δt in Fig. 6). This is followed by the estimate of I_t at the photo-capturing time t using the following equation:

$$I_t = I_\mathrm{L} + \frac{t - t_\mathrm{L}}{\Delta t}(I_\mathrm{R} - I_\mathrm{L}) \tag{4}$$

Table 1
Average values of performance metrics over five irCNN model runs applied to SD1 (validation performance).

Number of different backgrounds in images	MAE	MAPE (%)	R^2	NSE	KGE
100	1.07	7.67	0.77	0.75	0.87
200	0.73	5.27	0.88	0.87	0.87
300	0.62	4.47	0.91	0.91	0.90
400	0.60	4.28	0.91	0.91	0.91
500	0.57	4.10	0.92	0.92	0.92
600	0.56	4.01	0.91	0.91	0.96
700	0.54	3.85	0.93	0.93	0.96
800	0.54	3.81	0.94	0.94	0.96
900	0.54	3.80	0.94	0.94	0.95
1000	0.53	3.73	0.93	0.93	0.96
1100	0.52	3.65	0.94	0.94	0.96
1200	0.52	3.68	0.94	0.94	0.96

MAE: mean absolute error; MAPE: mean absolute percentage error; R^2: the coefficient of determination; NSE: Nash–Sutcliffe model efficiency; KGE: Kling–Gupta efficiency.

Table 2
Average values of performance metrics over five irCNN model runs applied to SD2 (validation performance).

Number of SIs	Selected SIs	MAE	MAPE (%)	R^2	NSE	KGE
2	SI = 10, 19	1.47	10.29	0.81	0.59	0.65
	SI = 13, 16	1.45	10.56	0.78	0.61	0.47
	SI = 14, 15	2.06	15.04	0.80	0.28	0.17
	Average	1.66	11.96	0.79	0.49	0.43
3	SI = 10, 15, 19	0.77	5.48	0.90	0.88	0.90
	SI = 11, 14, 17	0.84	5.71	0.88	0.86	0.81
	SI = 13, 14, 15	1.68	11.78	0.84	0.48	0.34
	Average	1.09	7.66	0.88	0.74	0.68
4	SI = 10, 13, 16, 19	0.62	4.30	0.92	0.91	0.96
	SI = 13, 14, 17, 18	0.94	7.44	0.87	0.82	0.71
	SI = 13, 14, 15, 16	1.30	9.50	0.88	0.69	0.50
	Average	0.95	7.08	0.89	0.81	0.72
5	SI = 10, 12, 15, 17, 19	0.54	3.81	0.94	0.93	0.98
	SI = 12, 14, 15, 17, 18	0.75	5.77	0.92	0.89	0.78
	SI = 13, 14, 15, 16, 17	1.07	8.29	0.90	0.77	0.59
	Average	0.79	5.96	0.92	0.86	0.78
6	SI = 10, 12, 14, 15, 17, 19	0.56	3.91	0.93	0.92	0.97
	SI = 11, 12, 14, 15, 16, 18	0.58	4.12	0.94	0.93	0.88
	SI = 12, 13, 14, 15, 16, 17	0.83	5.95	0.92	0.87	0.73
	Average	0.66	4.66	0.93	0.91	0.86
7	SI = 10, 12, 13, 15, 16, 18, 19	0.53	3.67	0.94	0.94	0.97
	SI = 11, 12, 14, 15, 17, 18, 19	0.56	4.05	0.93	0.93	0.93
	SI = 11, 12, 13, 14, 15, 16, 17	0.75	5.09	0.92	0.88	0.80
	Average	0.62	4.27	0.93	0.92	0.90
8	SI = 10, 11, 13, 14, 15, 16, 18, 19	0.53	3.69	0.94	0.94	0.96
	SI = 10, 12, 13, 14, 16, 17, 18, 19	0.53	3.76	0.94	0.94	0.97
	SI = 11, 12, 13, 14, 15, 16, 17, 18	0.60	4.32	0.93	0.92	0.87
	Average	0.56	3.92	0.94	0.93	0.93
9	SI = 10, 11, 12, 13, 14, 16, 17, 18, 19	0.51	3.62	0.94	0.94	0.95
	SI = 10, 11, 13, 14, 15, 16, 17, 18, 19	0.53	3.73	0.93	0.93	0.96
	SI = 10, 11, 12, 13, 14, 15, 16, 17, 18	0.57	3.91	0.93	0.92	0.92
	Average	0.54	3.75	0.94	0.93	0.94
10	SI = 10–19	0.54	3.89	0.94	0.94	0.95

where I_t is the rainfall intensity at the photo-capturing time t. I_L, I_R, t_L, and Δt are all illustrated in Fig. 6.

The smart phones took images of 11 rainfall events occurring between May and July 2020 at the locations shown in Fig. 5. This resulted in a total of 960 rainfall images with different backgrounds. Fig. 7 presents four examples of rainfall images with the intensity estimated using the method given in Fig. 6 and Eq. (4). A total of 768 (80%) of the above images are randomly selected to train the irCNN model; the remaining 192 (20%) images are used to validate the model performance.

3.3. *Real rainfall images from an* in situ *surveillance camera*

An *in situ* surveillance camera was installed in this study to capture rainfall videos, at the location shown in Fig. 5. Six rainfall events occurring in June and July 2020 were recorded by this *in situ* camera, and the videos were used as supplemental material. The rainfall videos are split into rainfall frames with 1 s resolution (i.e., still images) to enable the application of the proposed irCNN model. Rainfall intensity data are taken from the rainfall gauge shown in Fig. 5; the linear interpolation method described in

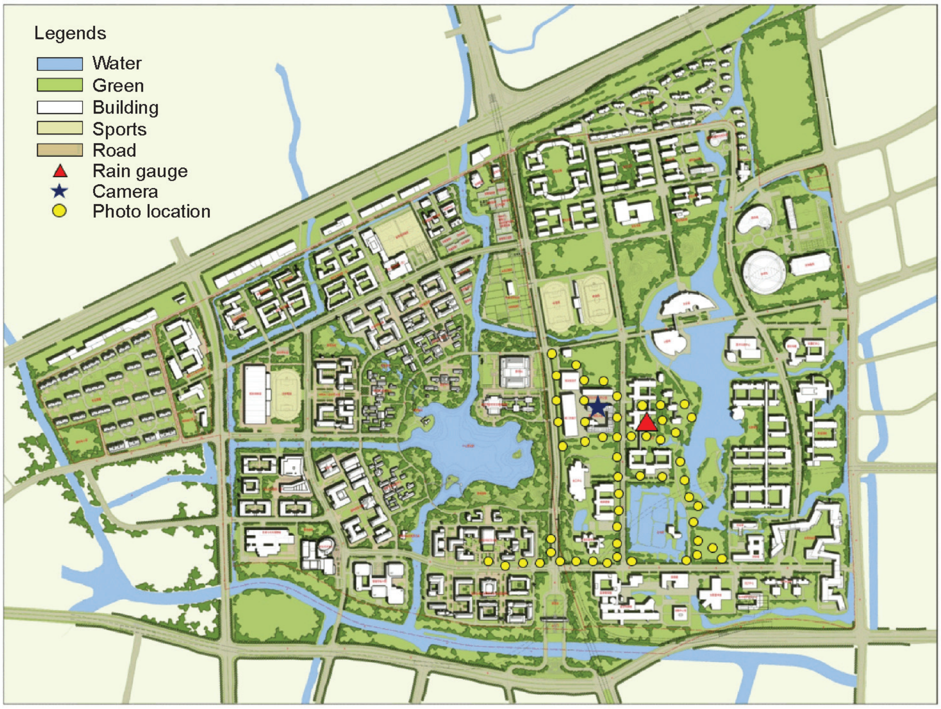

Fig. 5. Locations of the sensors (smart phones and cameras) used to collect rainfall images.

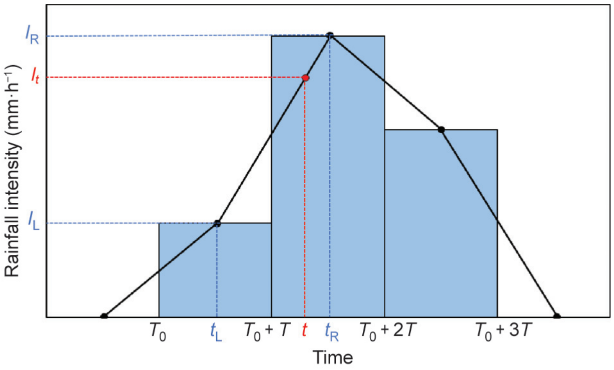

Fig. 6. Illustration of rainfall intensity estimation at any giving time t using the linear interpolation method. Black lines represent the linear interpolation results and blue histograms represent the recorded rainfall depth.

Fig. 6 and Eq. (4) is employed to assign a rainfall intensity to each video frame. A total of 7117 rainfall frames are produced from the camera videos based on the six rainfall events. In this study, 5694 (80%) frames are randomly selected to train the irCNN model, and the remaining 1423 (20%) frames are used to test the model's performance (this dataset is denoted as CD1). In addition, five of the six rain events are used to train the irCNN model, while the remaining rainfall event is utilized for model validation (this dataset is denoted as CD2).

4. IrCNN model training and validation

4.1. Model training

While various deep learning models have been successful in a range of applications, model training is often difficult due to the large number of model parameters involved [53]. In this study, the stochastic gradient descent (SGD) method is used to train the irCNN model, due to its previously demonstrated excellent efficiency and effectiveness [53]. Within the SGD method, the cyclical learning rate (CLR) approach is employed to speed up the training process. The details of the model training method can be found in the work of Ref. [53].

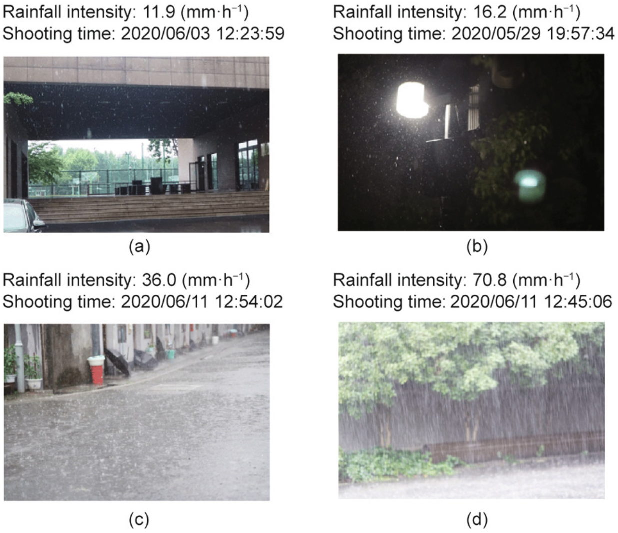
Rainfall intensity: 11.9 (mm·h^{-1})
Shooting time: 2020/06/03 12:23:59
(a)
Rainfall intensity: 16.2 (mm·h^{-1})
Shooting time: 2020/05/29 19:57:34
(b)
Rainfall intensity: 36.0 (mm·h^{-1})
Shooting time: 2020/06/11 12:54:02
(c)
Rainfall intensity: 70.8 (mm·h^{-1})
Shooting time: 2020/06/11 12:45:06
(d)

Fig. 7. Examples of rainfall images with estimated intensity from smart phones.

4.2. Performance metrics

Five metrics that have been widely used in the hydrology domain are used to measure the performance of irCNN models [54]. These are the mean absolute error (MAE), the mean absolute percentage error (MAPE), the coefficient of determination (R^2), the

Nash–Sutcliffe model efficiency (NSE), and the Kling–Gupta efficiency (KGE). The equations for the MAE and MAPE are as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|Y_i - \hat{Y}_i\right| \tag{5}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{Y_i - \hat{Y}_i}{Y_i}\right| \tag{6}$$

where n is the total number of data points, Y_i is the ith observation, and $\hat{Y}_i$ is the ith prediction. A lower value of MAE or MAPE indicates a better performance. The metrics of R^2, NSE, and KGE are to measure the goodness-of-fit of the model; the equations for these metrics are presented below:

$$R^2 = \frac{\left(\sum_{i=1}^{n}\left(Y_i - \widetilde{Y}\right)\left(Y_i - \overline{Y}\right)\right)^2}{\sum_{i=1}^{n}\left(Y_i - \widetilde{Y}\right)^2 \sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2} \tag{7}$$

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{n}\left(Y_i - \hat{Y}_l\right)^2}{\sum_{i=1}^{n}\left(Y_i - \overline{Y}\right)^2} \tag{8}$$

$$\mathrm{KGE} = 1 - \sqrt{(r-1)^2 + \left(\frac{\sigma_{\mathrm{pred}}}{\sigma_{\mathrm{obs}}} - 1\right)^2 + \left(\frac{\mu_{\mathrm{pred}}}{\mu_{\mathrm{obs}}} - 1\right)^2} \tag{9}$$

where $\overline{Y}$ is the mean of the observations; r is the linear correlation between the observations and the predictions; σ_{pred} and μ_{pred} are the standard deviation and the mean of the predictions, respectively; and σ_{obs} and μ_{obs} are the standard deviation and the mean of the observations, respectively. A higher value of R^2, NSE, or KGE indicates an overall better performance, with R^2, NSE, or KGE = 1 representing perfect model performance.

5. Results and discussions

5.1. Convergence and efficiency analysis

The proposed irCNN model was implemented in this study using the Python computer language. The implemented algorithm was run on a personal computer (PC) with an Intel Core i9-9820X at 3.3 GHz and 32 GB random access memory (RAM), with a NVIDIA RTX 2080Ti 11 GB graphic processing unit (GPU). It should be noted that the irCNN model was pretrained using the open-source dataset called ImageNet [50]. In other words, the convergence and efficiency analysis below were conditioned on the pretrained irCNN models.

Fig. 8(a) shows the convergence trajectories of the proposed irCNN model applied to the synthetic dataset SD1, where the minimization of the training loss is the objective function, as defined by Smith and Topin [53]. As shown in this figure, while different model runs may exhibit different convergence properties, they are all able to reach convergences between 10–50 training epochs. Furthermore, it is found that, although an irCNN model with a relatively low number of background images (i.e., low background diversity) tends to converge within a large number of training epochs, each training epoch requires a relatively low time budget. Using the computer configuration stated above, each irCNN model run applied to the synthetic dataset lasted between 10–20 min.

Fig. 8(b) presents the convergence trajectories of the irCNN model applied to the real rainfall images from smart phones. It can be seen that the number of training epochs needed for the real data is significantly larger than the number needed for the synthetic dataset. In addition, each training epoch for the former requires approximately 3 min, which is appreciably longer than that for the synthetic dataset. This is expected due to the noise that is present in the real rainfall images, which substantially increases the training difficulties. As shown in Fig. 8(b), all the irCNN model runs successfully converged within 3 h using the computer configuration previously stated. Similar observations can be made for the irCNN model applied to the synthetic dataset SD2, as well as the real rainfall images obtained by the *in situ* surveillance camera.

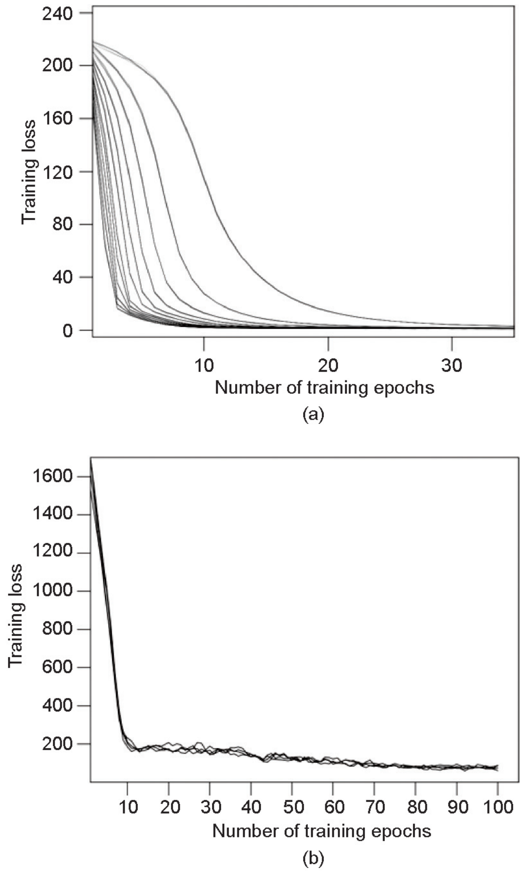

Fig. 8. Convergence trajectories of the proposed irCNN model runs: the convergence trajectories of the proposed model applied to (a) the synthetic dataset SD1 and (b) the real rainfall images from smart phones.

The time used to estimate the rainfall intensity using the trained irCNN model was recorded. Although it varied slightly for different input rainfall images, the irCNN model required 1–2 s to provide a rainfall intensity estimate for 100 images. This finding highlights the great potential of the irCNN model for providing real-time rainfall intensity once it has been trained using historical observations in urban areas.

5.2. IrCNN model performance on synthetic rainfall images

Table 1 shows the performance metric for the irCNN model applied to the synthetic datasets, where the metric values of the validation data are presented. It can be seen that, for each fixed number of background images in SD1, five different model runs with different randomly selected background images are performed, resulting in the averaged performance metric values shown in Table 1. It was notable that, for a fixed number of background images, different model runs showed a low variation in performance metric values (not shown here).

As shown in Table 1, the irCNN model performance is good overall (e.g., the average values of R^2, NSE, and KGE are all above 0.9) if sufficient background diversity can be guaranteed within the model training process. In addition, the performance of the irCNN model improves when the number of background images is increased from 100 to 600, followed by an overall similar model performance for further increases in the number of background images. In other words, the irCNN model can distinguish raindrops from the background images as long as a sufficient number of different background images are used for model training.

Based on the results shown in Table 1, we decided to use 800 fixed background images to enable the analysis of synthetic data in SD2—that is, to investigate the potential impacts of different rainfall scenarios on model performance. Table 2 presents the average performance metric values of the validation data over five model runs applied to each data subset in SD2. As shown in the table, the number of rainfall scenarios significantly affects model performance. For example, if the number of SI is 9, the average MAE, MAPE, R^2, NSE, and KGE of the irCNN model runs (for validation data) are 0.54, 3.75%, 0.94, 0.93, and 0.94, respectively. This represents a significantly improved performance when compared with the case that considered two or three different SI values, as shown in Table 2.

According to Table 2, for a fixed set of SI values, if the selected alternatives can cover a large span of the total options, the performance of the irCNN model improves. This finding indicates that the irCNN model may not be able to provide accurate estimates for scenarios with rainfall intensities beyond those provided in the training dataset. This limitation is typical for most machine learning methods, as they tend to perform much better at interpolating than extrapolating beyond the dataset used for their training. Based on the results shown in Table 2, it can be concluded that the diversity of rainfall scenarios and the span of rainfall intensities have a significant influence on model performance. This finding implies that a collection of a sufficiently large number of events with different rainfall intensities is critical to the performance of the irCNN model.

5.3. IrCNN model performance on real rainfall images captured by smart phones

Table 3 shows the values of the performance metrics based on the validation results of the irCNN model applied to the real rainfall images captured by smart phones. To enable a rigorous analysis, five runs with different randomly selected training data are performed; the results are given in Table 3. The table shows that, while the metric values can differ slightly over different runs, all are acceptable in practice to accurately determine the rainfall intensity. This result is reflected in the good average values of MAE, MAPE, R^2, NSE, and KGE achieved by the irCNN model simulations for a 3.79 mm·h^{-1} rainfall, which are 18.53%, 0.96, 0.95, and 0.91, respectively.

Fig. 9 depicts the predictions versus observations for the results of trial 3 shown in Table 3, with the red line representing perfect model performance. As shown in the figure, despite some variations, the irCNN model predictions match the rainfall intensity observations well overall. This result implies that the irCNN model can provide acceptable rainfall intensity estimates in practice, based on rainfall images captured by smart phones. While such estimates may not be as accurate as those from a ground rainfall station, they can be obtained at high temporal and spatial resolutions with a low associated cost.

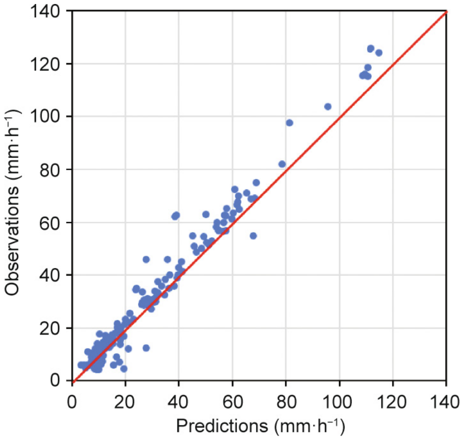

Fig. 9. Predictions vs observations based on the irCNN model applied to real rainfall images from smart phones (validation data).

5.4. IrCNN model performance on real rainfall images from a surveillance camera

A total of six rainfall events were recorded by the surveillance camera (details are given in Table 4); these videos were split into frames with a 1 s resolution to enable irCNN model application, as previously stated. Table 4 outlines the duration, average rainfall intensity, and maximum rainfall intensity of each rainfall event computed based on 1 min resolution records from the rain gauge. Following Ref. [14], rainfall data greater than 0.1 mm·min^{-1} (i.e., 6 mm·h^{-1}) are used for irCNN model development. It should be noted that storm burst events often occur in Hangzhou (the city where the surveillance camera was installed) between June and July; hence, the recorded events are mainly rainfall extremes with relatively short duration, as shown in Table 4. Such rainfall events are more likely to cause flash floods than average rainfall events; therefore, their spatiotemporal intensity values in an urban area are important for real-time flood defense (which is the focus of this paper). Nevertheless, future research should also validate the performance of the irCNN model in estimating rainfall intensity for average rainfall events (i.e., low-intensity events with a long duration).

Five different model runs with different randomly selected training data were performed; the validation results are given in Table 5. As shown in the table, the irCNN model can provide reasonably accurate rainfall intensity estimates based on real rainfall images from the surveillance camera, with the average values of MAE, MAPE, R^2, NSE, and KGE being 3.10 mm·h^{-1}, 16.54%, 0.92, 0.92, and 0.95, respectively. The irCNN model predictions versus observations for trial 4 (Table 5) are presented in Fig. 10. While

Table 3
Values of the performance metrics for the irCNN model runs applied to real rainfall images from smart phones (validation results).

Trial number	MAE (mm·h^{-1})	MAPE (%)	R^2	NSE	KGE
1	4.16	18.11	0.95	0.93	0.90
2	3.92	19.27	0.95	0.94	0.90
3	3.36	17.89	0.97	0.96	0.93
4	3.91	16.30	0.96	0.94	0.91
5	3.58	21.08	0.96	0.96	0.92
Average	3.79	18.53	0.96	0.95	0.91

Table 4
Details of six rainfall events recorded by the surveillance camera.

Rain event	Date	Duration (min)	Average rainfall intensity (mm·h^{-1})	Maximum rainfall intensity (mm·h^{-1})
1	2020/6/21	17	16.9	42.0
2	2020/6/21	69	19.0	66.0
3	2020/6/26	21	22.3	60.0
4	2020/7/10	12	11.0	36.0
5	2020/7/16	18	13.7	42.0
6	2020/7/26	33	23.6	60.0

Table 5
Values of the performance metrics for the irCNN models applied to real rainfall images from the surveillance camera (validation results).

Trial number	MAE (mm·h^{-1})	MAPE (%)	R^2	NSE	KGE
1	3.02	15.78	0.91	0.91	0.94
2	3.21	17.17	0.91	0.91	0.95
3	2.87	15.38	0.94	0.93	0.93
4	3.10	16.71	0.92	0.92	0.96
5	3.28	17.66	0.92	0.92	0.96
Average	3.10	16.54	0.92	0.92	0.95

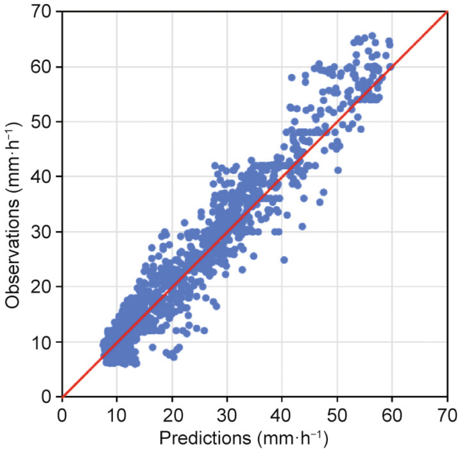

Fig. 10. Predictions vs observations based on the irCNN model applied to real rainfall images from the surveillance camera (validation data).

some variations can be observed, especially in the region with relatively high rainfall intensities, the irCNN model predictions match the observations well overall. It is observed that the performance of the irCNN model when applied to real rainfall images is deteriorated compared with the corresponding models developed using the synthetic dataset (Tables 1, 2, 3, and 5). This is because: ① The noise in real rainfall images is typically more complex than that in synthetic images due to the impact of the surrounding environment, such as the brightness or the weather conditions; and ② using a linear interpolation method (Fig. 6) for estimating the rainfall intensity at the image capture time inevitably induces errors. Still, the irCNN model exhibits a reasonable performance when handling real rainfall images, as demonstrated in Tables 3 and 5.

To further explore the performance of the irCNN model in predicting rainfall intensity based on images from a new rainfall event, five independent rainfall events are used for model training and the remaining independent rainfall event is used for model validation, with the results given in Table 6 and Fig. 11. As shown in Table 6, rainfall events 1 and 4 are selected for model validation, as ① the rainfall intensities of these two rainfall events are moderate compared with other events; and ② the rainfall duration of other events is relatively long, so they are used for model training (model training often needs a sufficient number of data points). As shown in Table 6 and Fig. 11, when trained and validated by independent rainfall events, the irCNN model performance is worse than when using randomly selected frames for model training (Table 5 and Fig. 10). For example, the average values of MAE, MAPE, R^2, NSE, and KGE of the irCNN model trained and validated using independent rainfall events are 3.78 mm·h^{-1}, 20.23%, 0.81, 0.76, and 0.87, respectively. This result shows a slightly deteriorated performance compared with the results from the model that was trained and validated using randomly selected rainfall images (Table 5). A similar observation can be made when comparing the results between Figs. 10 and 11.

The relative performance of the irCNN model when trained (and validated) using images from independent rainfall events or using randomly selected rainfall images is caused by the environmental variation (e.g., brightness and wind conditions) over different rainfall events (this study uses only one camera with a fixed angle to make videos). More specifically, the weather conditions during a single rainfall event can remain similar throughout the rainfall process, but can differ significantly among different rainfall events. Therefore, the use of images from independent rainfall events can increase the difficulty of model prediction. Still, in the worst case, the irCNN model prediction has an MAPE of 21.90%, which is still similar to the corresponding value presented by Jiang et al. [34] (an MAPE of 21.80%), who used a decomposition-based identification algorithm to estimate rainfall intensity. However, the trained irCNN model is significantly more computationally efficient than the method of Jiang et al. [34], as the proposed model takes approximately 1 s to estimate the rainfall intensity for 100

Table 6
Values of the performance metrics for the irCNN models applied to independent rainfall events with real rainfall images from the surveillance camera (validation results).

Rain event for model validation	Rain event for model training	MAE (mm·h^{-1})	MAPE (%)	R^2	NSE	KGE
1	2, 3, 4, 5, 6	2.40	18.55	0.93	0.93	0.94
4	1, 2, 3, 5, 6	4.35	21.90	0.69	0.60	0.80
Average	—	3.78	20.23	0.81	0.76	0.87

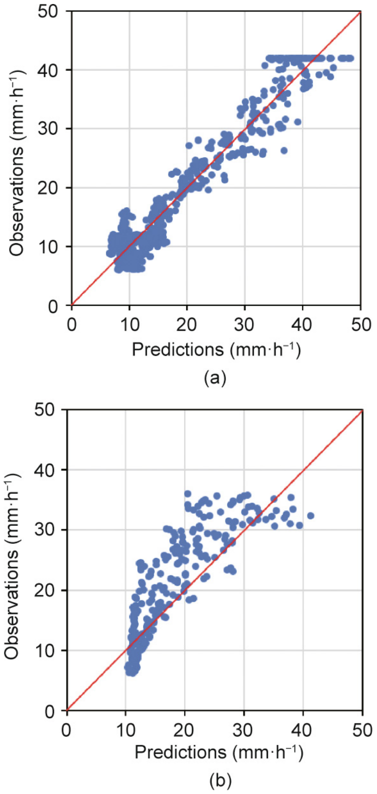

Fig. 11. Predictions vs observations based on the irCNN model applied to independent rainfall events with images from the surveillance camera (validation data).

images while that of Jiang et al. takes 26.4 s to perform the same task. This comparison highlights the great potential of the proposed irCNN model for near real-time flood risk management. In addition, the proposed irCNN can estimate rainfall intensity based on images (frames) not only from a surveillance camera, but also from other data sources such as smart phones. In comparison, the method of Jiang et al. [34] can only be used to estimate rainfall intensity based on rainfall videos from security cameras. However, it should be noted that, when many different cameras are used to collect rainfall images for the proposed irCNN model, the camera types and video-making angles may also affect model accuracy, in addition to the environmental conditions.

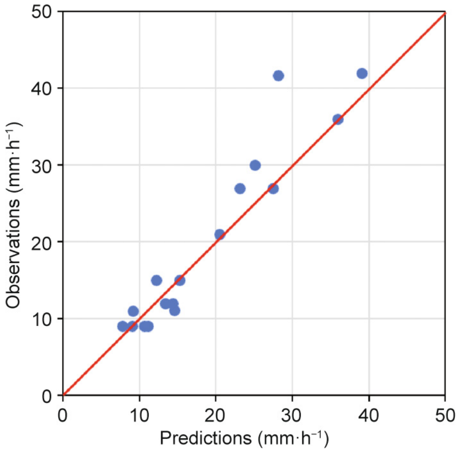

Fig. 12. Predictions vs observations (1 min time resolution) based on the irCNN model applied to independent rainfall events with images from the surveillance camera (validation data).

It should be acknowledged that while a linear interpolation method (Fig. 6) is used to approximate the rainfall intensity over different times based on the rain gauge records (1 min of accumulated rainfall depth), the real rainfall process may not be completely temporally linear with respect to intensity. To address this issue, the mean of the rainfall intensity is computed based on estimates from the irCNN model and is applied to all camera frames within 1 min. Using this approach, the rainfall intensity estimates of the two rainfall events (rainfall events 1 and 4 in Table 6) with a 1 min resolution are presented in Fig. 12. The MAE and MAPE values of these estimates are 2.55 mm·h^{-1} and 13.5%, which are significantly lower—that is, better than—the corresponding values in Table 6. This result indicates that using the mean rainfall intensity estimate—that is, a 1 min time resolution—improves the accuracy of the irCNN model. In engineering practice, a 1 min time resolution of the rainfall data is sufficient to enable urban real-time flooding management and operation [55].

6. Summary and conclusions

High-resolution spatiotemporal rainfall data in urban areas are fundamental to the real-time management (i.e., prediction, operation, and evacuation) of urban flooding. While many approaches are available to measure or predict rainfall intensity, including ground rainfall stations, weather radar, and satellite remote sensing, their rainfall measurements are either insufficient for the required spatiotemporal resolution or unsatisfactory in terms of accuracy. This paper proposed an image-based deep learning model to measure rainfall intensity with high spatiotemporal resolution. More specifically, a CNN model was developed (denoted as the irCNN model) for which images collected from existing dense sensors within rainfall events are the model inputs, and the corresponding rainfall intensity represents the model outputs.

Two different rainfall data types were used to explore the performance of the irCNN model in this study. Synthetic rainfall data were generated to systematically investigate the irCNN's ability in theoretically modeling rainfall intensity under different model development conditions such as different backgrounds and rainfall diversities in the training data. Real rainfall images captured by smart phones and a surveillance camera were then used to demonstrate the irCNN's practical utility. Based on the results, the main findings are as follows:

(1) The results based on synthetic rainfall data show that the irCNN model consistently provided an accurate rainfall estimate with an MAPE below 5.0% if sufficient background and rainfall event diversity were included in the training data. It was also found that the performance of the irCNN model was significantly affected by the background diversity of the images and by rainfall event diversity.

(2) The irCNN model successfully provided rainfall intensity estimates based on images captured by smart phones and a surveillance camera (i.e., rainfall videos), thereby demonstrating its great potential for engineering applications. The results based on real rainfall images show that the irCNN model provided rainfall estimates with an MAPE ranging between 13.5%–21.9% (with a mean of 16.5%). This average performance exceeds the corresponding accuracy (21.8% MAPE) of the decomposition-based identification algorithm [34], which is currently the state-of-the-art modeling technique. In addition, the proposed irCNN method was significantly more computationally efficient (about 20 times faster) than the decomposition-based identification algorithm [34]. Finally, the method of Jiang et al. [34] can only use rainfall videos to estimate rainfall intensity—that is, it cannot use still images to estimate rainfall intensity, unlike the irCNN model.

In summary, the proposed image-based deep learning model was demonstrated to be efficient and effective in acquiring urban rainfall data with high spatiotemporal resolution. The most important feature of the proposed irCNN model is its low cost in acquiring high spatiotemporal rainfall data in urban areas, as it uses existing sensors to collect rainfall images. We consider that the irCNN model provides a promising alternative to the other means that are currently available for measuring urban rainfall intensity. The high spatiotemporal data acquired by the model not only facilitate real-time urban flooding risk management, but also provide an opportunity to understand how the changing environment (i.e., due to climate change, urbanization, and the heat island effect) affects the local urban hydrologic process.

We acknowledge that the wide application of the proposed irCNN model can be challenging due to a number of factors. These include ① the availability of rainfall images from various sensors and the corresponding rainfall intensity values used for model training; ② the transmission efficiency of rainfall images from widely distributed sensors to the data center for processing the irCNN application in near real-time; and ③ the quality of the rainfall images under various environmental conditions (e.g., daytime, night, position of cameras under trees) and sensor conditions. Further research is required to address the aforementioned issues and improve the predictive capability of the irCNN model. In addition, the uncertainty associated with different aspects of the proposed method, as well as a comprehensive comparison over different rainfall measurement models (e.g., Jiang et al. [34]), need to be explored in the future. Another important future direction is to incorporate the data from ground rainfall stations into the proposed model framework, thereby further improving its accuracy in estimating rainfall intensity. While temporal and spatial corrections of rainfall intensities are difficult to quantify due to their variation over different storm events, their incorporation into the proposed model framework is likely to improve the irCNN model's predictive performance. Therefore, this challenge is worth exploring in the future.

Acknowledgments

This work is funded by the National Natural Science Foundation of China (51922096), and the Excellent Youth Natural Science Foundation of Zhejiang Province, China (LR19E080003). The author Dr. Huan-Feng Duan appreciates the support from the Hong Kong Research Grants Council (RGC) (15200719).

Compliance with ethics guidelines

Hang Yin, Feifei Zheng, Huan-Feng Duan, Dragan Savic, and Zoran Kapelan declare that they have no conflicts of interest or financial conflicts to disclose.

References

[1] Zheng F, Thibaud E, Leonard M, Westra S. Assessing the performance of the independence method in modeling spatial extreme rainfall. Water Resour Res 2015;51(9):7744–58.
[2] Gülbaz S, Kazezyılmaz-Alhan CM. Experimental investigation on hydrologic performance of lid with rainfall-watershed-bioretention system. J Hydrol Eng 2017;22(1):D4016003.
[3] Wahl T, Jain S, Bender J, Meyers SD, Luther ME. Increasing risk of compound flooding from storm surge and rainfall for major us cities. Nat Clim Chang 2015;5(12):1093–7.
[4] Zhou Q, Leng G, Su J, Ren Y. Comparison of urbanization and climate change impacts on urban flood volumes: importance of urban planning and drainage adaptation. Sci Total Environ 2019;658:24–33.
[5] Westra S, Fowler HJ, Evans JP, Alexander LV, Berg P, Johnson F, et al. Future changes to the intensity and frequency of short-duration extreme rainfall. Rev Geophys 2014;52(3):522–55.
[6] Archer DR, Fowler HJ. Characterising flash flood response to intense rainfall and impacts using historical information and gauged data in Britain. J Flood Risk Manag 2018;11:S121–33.
[7] Alves A, Vojinovic Z, Kapelan Z, Sanchez A, Gersonius B. Exploring trade-offs among the multiple benefits of green–blue–grey infrastructure for urban flood mitigation. Sci Total Environ 2020;703:134980.
[8] Morris ZB, Malone SM, Cohen AR, Weissburg MJ, Bras B. Impact of low-impact development technologies from an ecological perspective in different residential zones of the city of Atlanta, Georgia. Engineering 2018;4(2):194–9.
[9] Fletcher TD, Shuster W, Hunt WF, Ashley R, Butler D, Arthur S, et al. SUDS, LID, BMPS, WSUD and more–the evolution and application of terminology surrounding urban drainage. Urban Water J 2015;12(7):525–42.
[10] He M, Chen C, Zheng F, Chen Q, Zhang J, Yan H, et al. An efficient dynamic route optimization for urban flooding evacuation based on cellular automata. Comput Environ Urban Syst 2021;87:101622.
[11] Zhao Q, Yao Y, Yao W, Li Z. Real-time precise point positioning-based zenith tropospheric delay for precipitation forecasting. Sci Rep 2018;8(1):7939.
[12] Lee O, Sim I, Kim S. Application of the non-stationary peak-over-threshold methods for deriving rainfall extremes from temperature projections. J Hydrol 2020;585:124318.
[13] Yao L, Wei W, Chen L. How does imperviousness impact the urban rainfall–runoff process under various storm cases? Ecol Indic 2016;60:893–905.
[14] Berg P, Moseley C, Haerter JO. Strong increase in convective precipitation in response to higher temperatures. Nat Geosci 2013;6(3):181–5.
[15] Wasko C, Sharma A. Steeper temporal distribution of rain intensity at higher temperatures within Australian storms. Nat Geosci 2015;8(7):527–9.
[16] Muller CL, Chapman L, Johnston S, Kidd C, Illingworth S, Foody G, et al. Crowdsourcing for climate and atmospheric sciences: current status and future potential. Int J Climatol 2015;35(11):3185–203.
[17] Moron V, Barbero R, Fowler HJ, Mishra V. Storm types in India: linking rainfall duration, spatial extent and intensity. Philos Trans Royal Soc Math Phys Eng Sci 2021;379(2195):20200137.
[18] Breinl K, Müller-Thomy H, Blöschl G. Space–time characteristics of areal reduction factors and rainfall processes. J Hydrometeorol 2020;21(4): 671–89.
[19] Berkhahn S, Fuchs L, Neuweiler I. An ensemble neural network model for real-time prediction of urban floods. J Hydrol 2019;575:743–54.
[20] Yang L, Tian F, Niyogi D. A need to revisit hydrologic responses to urbanization by incorporating the feedback on spatial rainfall patterns. Urban Clim 2015;12:128–40.
[21] Jamali B, Bach PM, Deletic A. Rainwater harvesting for urban flood management–an integrated modelling framework. Water Res 2020;171:115372.
[22] Xia X, Liang Q, Ming X, Hou J. An efficient and stable hydrodynamic model with novel source term discretization schemes for overland flow and flood simulations. Water Resour Res 2017;53(5):3730–59.
[23] Powers JG, Klemp JB, Skamarock WC, Davis CA, Dudhia J, Gill DO, et al. The weather research and forecasting model: overview, system efforts, and future directions. Bull Am Meteorol Soc 2017;98(8):1717–37.
[24] Shen M, Chen J, Zhuan M, Chen H, Xu CY, Xiong L. Estimating uncertainty and its temporal variation related to global climate models in quantifying climate change impacts on hydrology. J Hydrol 2018;556:10–24.
[25] Lazri M, Labadi K, Brucker JM, Ameur S. Improving satellite rainfall estimation from msg data in northern Algeria by using a multi-classifier model based on machine learning. J Hydrol. 2020;584:124705.
[26] Villarini G, Mandapaka PV, Krajewski WF, Moore RJ. Rainfall and sampling uncertainties: a rain gauge perspective. J Geophys Res Atmos 2008;113(D11): D11102.
[27] Thorndahl S, Einfalt T, Willems P, Nielsen JE, ten Veldhuis MC, Arnbjerg-Nielsen K, et al. Weather radar rainfall data in urban hydrology. Hydrol Earth Syst Sci 2017;21(3):1359–80.
[28] Yong B, Hong Y, Ren LL, Gourley JJ, Huffman GJ, Chen X, et al. Assessment of evolving TRMM-based multisatellite real-time precipitation estimation methods and their impacts on hydrologic prediction in a high latitude basin. J Geophys Res Atmos 2012;117(D9):D09108.
[29] Tan ML, Yang X. Effect of rainfall station density, distribution and missing values on swat outputs in tropical region. J Hydrol 2020;584:124660.
[30] Voormansik T, Cremonini R, Post P, Moisseev D. Evaluation of the dual-polarization weather radar quantitative precipitation estimation using long-term datasets. Hydrol Earth Syst Sci 2021;25(3):1245–58.
[31] Zorzetto E, Marani M. Downscaling of rainfall extremes from satellite observations. Water Resour Res 2019;55(1):156–74.
[32] Zheng F, Tao R, Maier HR, See L, Savic D, Zhang T, et al. Crowdsourcing methods for data collection in geophysics: state of the art, issues, and future directions. Rev Geophys 2018;56(4):698–740.
[33] Hut R, de Jong S, van de Giesen N. Using umbrellas as mobile rain gauges: prototype demonstration. EGU Gen Assem Conf Abstr 2014:16418.
[34] Jiang S, Babovic V, Zheng Y, Xiong J. Advancing opportunistic sensing in hydrology: a novel approach to measuring rainfall with ordinary surveillance cameras. Water Resour Res 2019;55(4):3004–27.
[35] Yuan Q, Shen H, Li T, Li Z, Li S, Jiang Y, et al. Deep learning in environmental remote sensing: Achievements and challenges. Remote Sens Environ 2020;241:111716.
[36] Reichstein M, Camps-Valls G, Stevens B, Jung M, Denzler J, Carvalhais N, et al. Deep learning and process understanding for data-driven Earth system science. Nature 2019;566(7743):195–204.

[37] Ham YG, Kim JH, Luo JJ. Deep learning for multi-year ENSO forecasts. Nature 2019;573(7775):568–72.

[38] Pyo J, Park LJ, Pachepsky Y, Baek SS, Kim K, Cho KH. Using convolutional neural network for predicting cyanobacteria concentrations in river water. Water Res 2020;186:116349.

[39] Raissi M, Yazdani A, Karniadakis GE. Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. Science 2020;367(6481): 1026–30.

[40] Guo Z, Leitão JP, Simões NE, Moosavi V. Data-driven flood emulation: speeding up urban flood predictions by deep convolutional neural networks. J Flood Risk Manag 2021;14(1):e12684.

[41] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.

[42] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. Proc IEEE Conf Comput Vis Pattern Recognit 2016:770–8.

[43] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998;86(11):2278–324.

[44] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 2012;25:1097–105.

[45] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. 2014. arXiv:1409.1556.

[46] Rawat W, Wang Z. Deep convolutional neural networks for image classification: a comprehensive review. Neural Comput 2017;29(9):2352–449.

[47] Aloysius N, Geetha MA. A review on deep convolutional neural networks. In: Proceedings of 2017 International Conference on Communication and Signal Processing (ICCSP); 2017 Apr 6–8; Chennai, India. New York City: IEEE; 2017.

[48] Canziani A, Paszke A, Culurciello E. An analysis of deep neural network models for practical applications. 2016. arXiv:1605.07678.

[49] Gupta A, Matta P, Pant B. Graph neural network: Current state of art, challenges and applications. Mater Today Proc 2021;46:10927–32.

[50] Deng J, Dong W, Socher R, Li LJ, Li K, Li FF. Imagenet: a large-scale hierarchical image database. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2009 Jun 20–25; Miami, FL, USA. New York City: IEEE; 2009. p. 248–55.

[51] Zhang H, Patel VM. Density-aware single image de-raining using a multi-stream dense network. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition; 2018 Jun 19–21; Salt Lake City, UT, USA. New York City: IEEE; 2018. p. 695–704.

[52] How to add rain to a photo with photoshop [Internet]. San Jose: Photoshop Essentials.com; 2021 [cited 2021 Jul 8]. Available from: https://www.photoshopessentials.com/photo-effects/photoshop-weather-effects-rain/.

[53] Smith LN, Topin N. Super-convergence: very fast training of neural networks using large learning rates. In: Proceedings of Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications; 2019 Apr 14–18; Baltimore, MD, USA. Bellingham: Society of Photo-Optical Instrumentation Engineers; 2019. p. 2019.

[54] Mu L, Zheng F, Tao R, Zhang Q, Kapelan Z. Hourly and daily urban water demand predictions using a long short-term memory based model. J Water Resour Plan Manage 2020;146(9):05020017.

[55] Wang C, Hou J, Miller D, Brown I, Jiang Y. Flood risk management in sponge cities: the role of integrated simulation and 3D visualization. Int J Disaster Risk Reduct 2019;39:101139.