Received: 28 April 2020 | Revised: 3 October 2020 | Accepted: 18 November 2020

DOI: 10.1111/jfr3.12684

ORIGINAL ARTICLE

CIWEM Chartered Institution of Water and Environmental Management | Journal of Flood Risk Management | WILEY

Data-driven flood emulation: Speeding up urban flood predictions by deep convolutional neural networks

Zifeng Guo[1] | **João P. Leitão**[2] | **Nuno E. Simões**[3] | **Vahid Moosavi**[1]

[1]Chair of Digital Architectonics, Institute of Technology in Architecture (ITA), Department of Architecture, Swiss Federal Institute of Technology Zurich (ETHZ), Zürich, Switzerland

[2]Department of Urban Water Management, Swiss Federal Institute of Aquatic Science and Technology (Eawag), Dübendorf, Switzerland

[3]INESC Coimbra, Department of Civil Engineering, University of Coimbra, Coimbra, Portugal

Correspondence
Zifeng Guo, Chair of Digital Architectonics, Institute of Technology in Architecture (ITA), Department of Architecture, Swiss Federal Institute of Technology Zurich (ETHZ), Stefano-Franscini-Platz 1, Zürich, 8093, Switzerland.
Email: guo@arch.ethz.ch

Abstract

Computational complexity has been the bottleneck for applying physically based simulations in large urban areas with high spatial resolution for efficient and systematic flooding analyses and risk assessment. To overcome the issue of long computational time and accelerate the prediction process, this paper proposes that the prediction of maximum water depth can be considered an image-to-image translation problem in which water depth rasters are generated using the information learned from data instead of by conducting simulations. The proposed data-driven urban pluvial flood approach is based on a deep convolutional neural network trained using flood simulation data obtained from three catchments and 18 hyetographs. Multiple tests to assess the accuracy and validity of the proposed approach were conducted with both design and real hyetographs. The results show that flood prediction based on neural networks use only 0.5% of the time compared with that of physically based models, with promising accuracy and generalizability. The proposed neural network can also potentially be applied to different but relevant problems, including flood analysis for flood-safe urban layout planning.

KEYWORDS

convolutional neural network, data-driven emulation, fast water depth prediction, flood modelling

1 | INTRODUCTION

The combination of rapid urbanization and the rainfall intensity increase due to climate change is posing great challenges for flood risk management (Plate, 2002). Fast flood prediction methods are required to conduct systematic analyses and investigations of different urban planning and climate change scenarios (Zheng, Thibaud, Leonard, & Westra, 2015). Furthermore, if rapid urban pluvial flood predictors are integrated with high temporal resolution online rainfall forecast services, it will be possible to inform citizens of likely urban pluvial flooding in advance, so that precautionary measures can be taken.

1.1 | Rapid flood modelling

The current bottleneck for rapid urban pluvial flood analyses is the long computational time required by physically based simulation models. The problem becomes extremely significant for large simulation areas with high spatial resolutions (small raster grid size). Solutions for

solving this issue have been previously identified, which include reducing the dimensionality of flood representation (e.g., Samuels, 1990) or neglecting the inertial and advection terms of the momentum equation (e.g., Bates, Horritt, & Fewtrell, 2010; Bradbrook, Lane, Waller, & Bates, 2004; Chen, Djordjevic, Leandro, & Savic, 2007). However, according to Teng et al. (2017), physically based models are still not suitable for areas larger than 1,000 km^2 with a raster grid size smaller than 10 m. In urban areas, the required raster grid size should be between 1 m and 5 m to capture the urban features (Fewtrell, Bates, Horritt, & Hunter, 2008; Leitão, Boonya-Aroonnet, Prodanović, & Maksimović, 2009).

Non-physically based models, on the other hand, are based on simplified hydrological concepts such as topographic depressions (L'homme et al., 2008) or the transition rules of cellular automata (e.g., Ghimire et al., 2013; Guidolin et al., 2016). These methods can be implemented using parallel computing techniques to significantly improve the computational speed. Recently, Jamali, Bach, Cunningham, and Deletic (2019) have shown that the cellular automata models can predict maximum water depths of large urban areas in a few minutes. But the main drawback lies in its sensitivity to simulation time steps and raster grid size. Reducing the time step or grid size still causes a significant increase in simulation time. Non-physically based approaches have become significantly faster than physically based approaches, however, they are still considered not fast enough for applications that require a considerable number of simulations, such as flood-driven optimizations and planning. Also, it is known that the computational time is non-linearly related with the raster resolutions. Doubling the resolution may cause ten times more simulation times (Guidolin et al., 2016).

1.2 | Data-driven methods for flow and flood related problems

Recently, solving physics-related problems with machine learning techniques has become a research field receiving growing attention due to its computational efficiency and the ability to learn and generalize from data (e.g., Greydanus, Dzamba, & Yosinski, 2019). This direction is also called "surrogate modelling" because the machine learning models act as the "surrogate" of the physically based simulations hypothesising that the models will learn the target system regardless of the actual physical process if sufficient data are provided. Making surrogate models for flow and flood related problems has been investigated using different methods. The noticeable early attempts can be found from the field of computer graphics. For example, regression forest was used to learn the dynamic process of particle-based simulations for real-time animation and layout planning (Feng, Yu, Yeung, Yin, & Zhou, 2016; Ladický, Jeong, Solenthaler, Pollefeys, & Gross, 2015). Later, a fully-connected neural network was proposed by Mustafa et al. (2018) for flood-driven urban planning. However, the approach had a main drawback that the training data for the neural network was the input parameters of a terrain generator rather than raw elevation data, and consequently the results cannot be extended to other scenarios. Recently, this issue was handled by introducing a prior terrain feature extraction process (e.g., Leitão, Zaghloul, & Moosavi, 2018; Zaghloul, 2017).

Using artificial neural networks (or neural networks) for flood prediction and susceptibility mapping has recently become more frequent due to (a) neural networks were shown to be able to approximate any strong non-linear correlation and (b) recurrent neural networks (Schuster & Paliwal, 1997), a neural network structure which stacks multiple layers dynamically, were found effective for processing time series data. Recurrent neural networks were used to forecast long-term water level for individual locations from corresponding inputs of rainfall intensity (e.g., Chang, Chen, Lu, Huang, & Chang, 2014; Gude, Corns, & Long, 2020), or combined with self-organizing maps to estimate maximum water depths (Kim & Han, 2020a). Fully-connected neural networks were used to estimate water levels and flow velocities at specific coordinates based on statistical and topographical inputs such as slope, aspect and curvatures (e.g., Bui et al., 2020; Kim & Han, 2020b). However, a main drawback of fully-connected neural networks is the exponential growth of parameters on large inputs (LeCun, Bengio, & Hinton, 2015), which cause significant challenges when trying to utilize spatial information from adjacent pixels (raster cells) for two-dimensional simulations. A workaround of this problem was shown by Berkhahn, Fuchs, and Neuweiler (2019) through a combination of pruning zero-cells from the neural network inputs and using multiple small neural networks that are responsible for different local areas instead of one large network that predicts the entire catchment area.

1.3 | Data-driven flow and flood modelling using convolutional neural networks

Despite the variety of available algorithms, convolutional neural networks (CNNs) in particular, have received strong attention due to their ability to (a) process raw data in image format and (b) utilize spatial information

from adjacent pixels with many fewer parameters compared to other type of neural networks. Beyond its original goal for image classification and object detections in computer graphics, CNNs have been used to solve flow-related problems in a way that the boundary conditions were discretized into images (rasters) with multiple image channels. For example, the velocity and pressure fields of steady inlet air flow around objects of interest were inferred by CNNs from discretized inputs (e.g., Guo, Li, & Iorio, 2016; Thuerey, Weißenow, Prantl, & Hu, 2020; Tompson, Schlachter, Sprechmann, & Perlin, 2017). Chu and Thuerey (2017) used weight-sharing CNNs (multiple CNN instances sharing the same trainable parameters) to learn the feature descriptor of smoke flows for synthesizing high-resolution predictions from low-resolution simulations. Hennigh (2017) predicted the flow dynamics by decoding the compressed initial conditions through a series of weight-sharing CNNs that stack on after each other. Gao, Sun, and Wang (2020) used CNNs for handling inputs with irregular boundaries using elliptic coordinate transformations.

The use of CNNs for flood-related problems, especially for urban pluvial flood predictions, has not received much attention. This may be justified by, first, in contrast to the fluid dynamic experiments focusing on objects that can be rasterized to a relatively low resolution without significant information loss (e.g., 256×256 pixels), flood prediction is usually conducted in large catchment areas that require sufficient spatial resolution to be preserved. Therefore, building the CNNs for flood prediction typically require large input sizes and thus more layers, creating difficulties in the training step of CNNs. Second, for the purpose of flood prediction, training CNNs requires a large dataset in raster format. Considering the scale and spatial and temporal resolutions of typical flood simulation tasks, it is computationally and memory expensive to prepare a large dataset. Lacking proper datasets has been the main obstacle for data-driven flood modelling and prediction studies. Nevertheless, a few recent studies have shown that CNNs are promising for solving flood-related problems with corresponding data available. For example, CNNs can be effective for flood extent mapping using aerial or street view imagery (e.g., Gebrehiwot, Hashemi-Beni, Thompson, Kordjamshidi, & Langan, 2019; Moy de Vitry, Kramer, Wegner, & Leitão, 2019). These methods can potentially be used to replace physically based simulators for massive data collection. Wang, Fang, Hong, and Peng (2020) adopted CNNs for flood susceptibility mapping from multiple topographic features that are derived from the raw elevation data. Based on the input information, the model predicts the flood sensitivity as five output categories.

2 | PROBLEM STATEMENT

The above review has suggested that data-driven methods are promising for addressing physics-related problems, including urban pluvial flood prediction. However, most previous flood-related studies using artificial neural networks did not utilize the spatial information of catchment areas. Among the few existing studies, CNNs were used for classifications (e.g., flood susceptibility mapping and flood extent mapping) rather than numerical predictions. Therefore, we propose to use CNNs for rapid data-driven flood predictions, specifically, to predict the maximum water depth from the inputs of both elevation rasters and hyetographs. Our main contributions include (a) investigating CNNs as an end-to-end method for flood predictions, which means the model directly predicts the water depth values from the raw inputs; (b) introducing a CNN architecture that combines both spatial inputs (rasters) and vector inputs (hyetographs); and (c) providing a systematic pipeline that can potentially be used to address other relevant problems.

Our study currently focuses on the prediction of maximum water depth and neglects the temporal dynamic of flooding. The maximum water depth reflects the worst flooding cases and is critical for urban planning, risk management, and damage assessment (Jamali et al., 2018). The prediction of maximum water depth can be further divided into three cases: (a) different storms in the same catchment; (b) same storm in different catchments, and (c) considering both the rainfall and catchment variations. As a first step, we would focus on the first case and investigate the ability of generalization of CNNs on different rainfall inputs in specific catchments. The present work does not include flow velocity, but other studies have shown that it can be done by adding extra image channels to the output, and the predictions between different image channels are independent (e.g., Hennigh, 2017; Thuerey et al., 2020).

3 | PROPOSED APPROACH

3.1 | Framework

Our urban pluvial flooding approach proposes a CNN model as the main prediction mechanism. The key idea is that the flood prediction task can be regarded as an image-to-image translation problem where both the input and output are images (rasters) of different data. The proposed CNN model operates on the patches rather than the entire catchment area. The reasons for working with patches includes: (a) patches from different locations may share similarities that help the CNN learn and

generalize; (b) the patch-based method is an effective way to produce more training data from a limited number of physically-based simulation results; and (c) patches can effectively reduce the input size of the CNN, easing the CNN training challenges.

The pipeline of the approach can be described as four main steps, as presented in Figure 1. First, flood simulations of several design storms in different catchment areas are conducted to produce a flood (maximum water depth) dataset. The obtained water depth results are split into training and test data sets according to their corresponding hyetographs. Second, the elevation data of the catchment areas are pre-processed into images with multiple image channels. Each channel corresponds to a different surface feature. The hyetographs of the design storms are represented as vectors in which each dimension corresponds to the average rainfall intensity in a 5-minute interval. Third, patch locations are randomly generated within the catchment area. The patch size equals the input size of the CNN. The patches are used to train the CNN model by supervised learning, for which the ground truths are the water depth patches, and the inputs are the corresponding hyetographs and terrain patches. Finally, after the training, the flood predictions of new storms are performed, but the locations of patches are sampled from a grid in order to minimize the number of necessary patches. The obtained maximum water depth patches are assembled as the final output.

3.2 | Catchment representation

Five terrain surface features are included in the terrain image for catchment representation: elevation, slope, aspect, curvature and mask. The slope is defined as the magnitude of the gradient vector at each raster cell, representing the maximum rate of change in value from the centre cell to its neighbours and reflecting the steepness of the terrain and the overall movement of the water. The aspect identifies the direction of the water flow at each raster cell and is the directional component of the gradient vector. The curvature is defined as profile curvature which describes the acceleration and deceleration of the flow, and plan curvature which describes the convergence and divergence of the flow. Our approach uses the difference between these two curvatures (De Smith, Goodchild, & Longley, 2007) for simplicity. The mask is a binary image that indicates the catchment areas and no-data areas as 1 and − 1, respectively. The above features are rescaled linearly to the range of [−1, 1] and concatenated as a multichannel image.

3.3 | CNN-based prediction model

As shown in Figure 2, the prediction network consists of a convolutional autoencoder (the main network) and a feedforward fully connected neural network (the sub-network) that attaches to the main network's latent layer. The main network and sub-networks process the terrain and hyetograph data, respectively. After the latent layer, the main network decodes the combined data and predicts the water depth values (metres). This type of neural network that processes data in different formats is usually called a joint model (Ngiam et al., 2011).

The encoder of the main network is a chain of convolutional modules that consists of three convolutional layers and one pooling layer. The decoder is a chain of up-sampling modules that contain one deconvolution layer (or formally called transposed convolution layer) followed by two convolutional layers. The dimensions of the input and output are $256 \times 256 \times 5$ and $256 \times 256 \times 1$, respectively (height × width × number of features). The sub-network consists of one fully connected layer and one reshape layer. The size of the fully connected layer is 4,096 for convenient concatenation to the main network. The kernel sizes are 3×3 for all the convolutional layers and 2×2 for all the pooling and deconvolution layers. We use a small kernel size to preserve the thin structure of the terrain and deep layers to

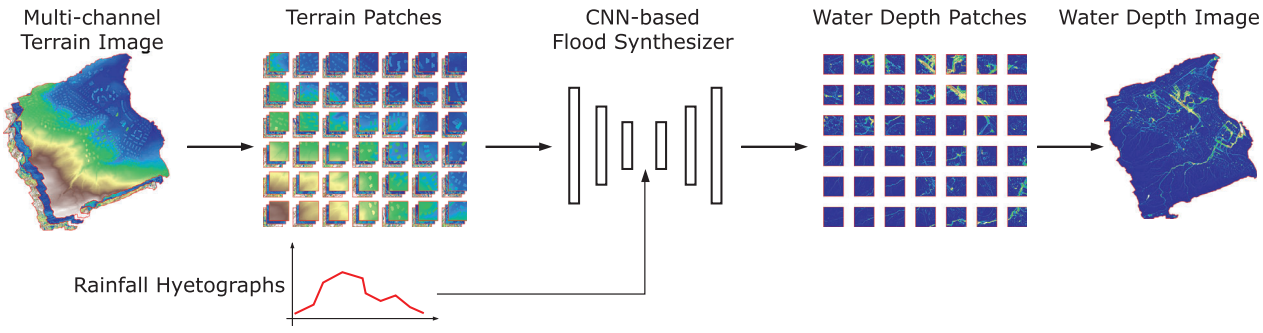

FIGURE 1 Data-driven flood emulation framework

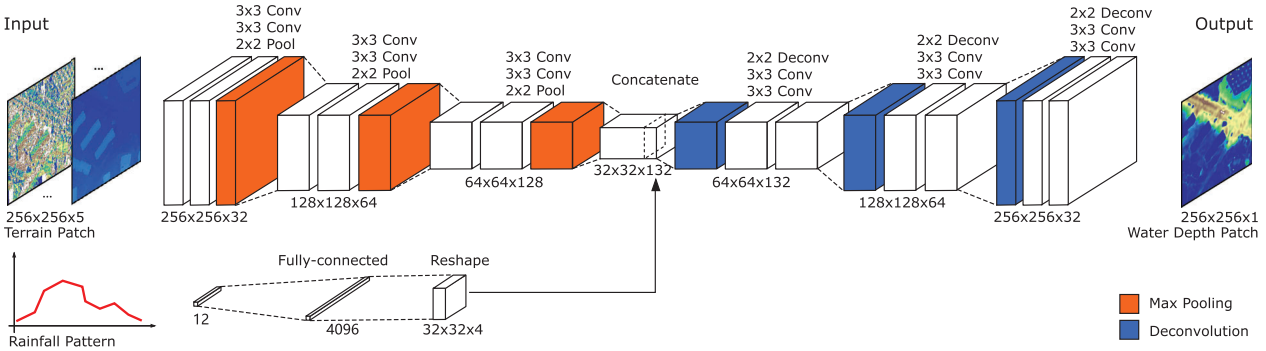

FIGURE 2 The flood prediction network

extend the receptive field (Luo, Li, Urtasun, & Zemel, 2016). The activation functions for all the layers are Leaky-ReLU (Maas, Hannun, & Ng, 2013) to avoid the "vanishing gradient problem" (Hochreiter, 1998) for sigmoid units and the dead neuron problem for rectified linear units caused by bad weight initialization (Nair & Hinton, 2010).

Water simulation results usually contain more no-water and shallow-water areas than deep-water areas, meaning that the dataset is imbalanced and can lower the accuracy of deep-water areas. Therefore, a weighted mean squared error is proposed for the loss function instead of the standard mean squared error. The definition of the loss is given as:

$$\frac{1}{n}\sum e^{y+c}(y-\hat{y})^2$$

where the loss weights are calculated by the exponentiation of the simulated water depth y plus the constant c; $\hat{y}$ is the predicted water depth and n is the number of samples. We found that with a larger c, the model tends to underestimate in deep-water areas. For all the tests in this paper, we use $c = -1$.

3.4 | Aggregating cell values from patches

The final water depth prediction is aggregated from the output water depth patches. As already mentioned, the patch locations (centre points) are determined by an orthogonal grid. The grid size is user-specified and should not be greater than the patch size. When the grid size equals the patch size, the adjacent patches touch each other's boundary without overlaps. Smaller grid sizes lead to more patches that overlap with each other, which provides redundant predictions for the overlapping areas to reduce outliers but increases the computational time. Choosing the grid size is a balance between time and accuracy. Our study investigated four patch aggregation options: patches without overlaps, and patches with overlaps aggregated by mean, median, or maximum values, respectively. The grid size for sampling prediction patches is 128 (half of the patch size). All results presented below are obtained using mean values unless mentioned otherwise.

4 | EXPERIMENTAL SETUP

We applied our framework to three different catchment areas located in Luzern and Zurich, Switzerland, and Coimbra, Portugal, using rasters with a grid size of 1 m. The CNN models are trained separately for each catchment. The framework was implemented in Python using TensorFlow 1.10 (Abadi et al., 2016). All the processes, including simulation, training, and validation, were performed with Graphics Processing Unit (GPU) parallel computing acceleration.

4.1 | Training data

To prepare the training data for the experiment, 18 one-hour rainfall events were created based on return periods of 2, 5, 10, 20, 50 and 100 years. Each return period corresponds to three different events. These events were randomly labelled as training and test sets except that each return period only occurs once in the test set (Figure 3). The same 18 events were used for the flood simulations in three catchment areas. The CNNs were trained using only the training set and were evaluated using the test set.

The simulations were conducted using the CADDIES cellular-automata flood model (Guidolin et al., 2016) for the Zurich and Luzern catchments, and Infoworks ICM software (Innovyze, 2019) for the Coimbra catchment.

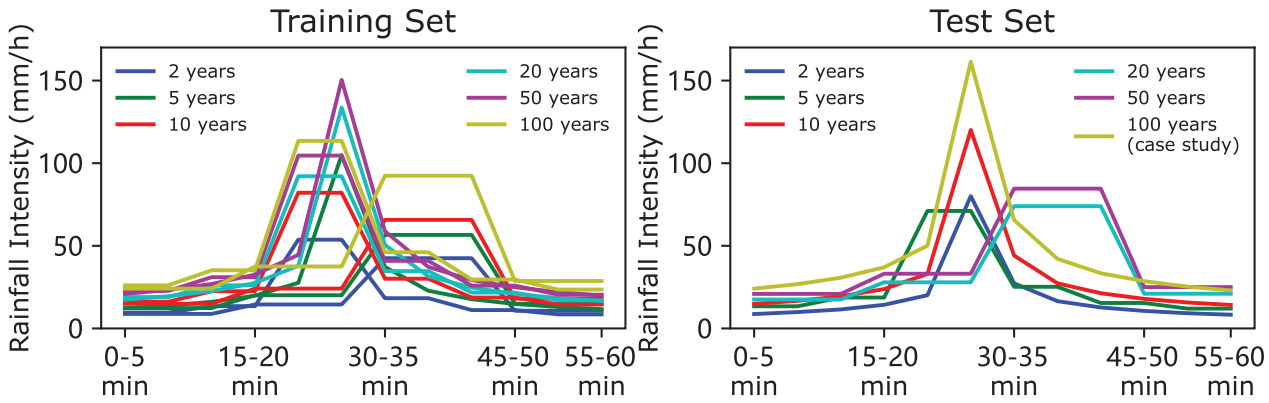

FIGURE 3 Hyetographs used for simulations

CADDIES is a cellular-based surface flood model, whereas Infoworks ICM is a physically based model that can consider coupled pipe/surface flow in urban areas. All the simulations were performed using rasters with a grid size of 1 m and a minimum time step of 0.01 s. Bruwier et al. (2018) have suggested that the 90th percentile should be used for analysing simulation results in order to represent the reality better. However, as we aimed to replicate the output of a flood simulator, we kept the raw simulation outputs without further postprocessing.

The training data were the hyetograph vectors and the patches sampled from the terrain elevation data and the water depth simulations. For each catchment area, 10,000 patch locations were randomly sampled. As there are 18 rainfall events created for the dataset, each patch location corresponds to 1 terrain patch and 18 water depth patches. The ground truth data are the water depth patches. One flood prediction model was trained for each catchment area in order to study the generalizability of CNN on different hyetograph inputs. We used identical meta parameters to train all models. Specifically, we used the Adam optimizer (Kingma & Ba, 2014) for 200 epochs, with a batch size of 32 and a fixed learning rate of 0.0001.

4.2 | Evaluation and validation

The performance of the proposed model was evaluated based on computational time, prediction accuracy and the ability of generalization on hyetographs. The computational times were measured by repeating the prediction process and calculating the average time. The time for necessary pre-processing (e.g., calculating the terrain features) is also reported. In addition, both the accuracy and computational time of different aggregation methods were analysed to discuss the trade-offs between speed and accuracy.

The accuracy was assessed by the mean absolute error (MAE) and the 2D histogram between all the raster cells of the predicted and simulated water depths. The MAE is defined as $\frac{1}{n}\sum_i^n |\hat{y}_i - y_i|$, where $\hat{y}_i$ and y_i are the i-th predicted and simulated raster cells, respectively. We used the MAE to assess the accuracy of the results produced by different meta parameters, such as the patch aggregation methods and the grid size for patch sampling. The 2D histogram is a plot in which the pixel at row i and column j represents the number of water depth raster cells that are y_i m by prediction and y_j m by simulation. The plot can also be used to show the simulation-error relation by replacing y_i to errors. The histograms serve as the alternative to scatter plots for better readability. In addition to these two assessment methods, the local performance in different areas such as upstream, downstream and depressions, and the spatial distribution of errors $\Delta y = \hat{y}_i - y_i$ and relative error $\delta y_i = \Delta y_i / y_i$ are also reported.

The trained models were also validated using real rainfall events[1] that were not included in the training data to further investigate the generalizability of our flood prediction model. Real rainfall events that were less than 70 minutes were selected, clipped to 60 minutes, and resampled to 12-dimensional vectors. The accuracies were reported using histograms as well as spatial plots.

5 | RESULTS

In this section, we present the result of comparing different patch aggregation methods as well as the detailed analysis of the best patch aggregation method. The latter consists of the accuracy analysis on the heaviest rainfall event (100-year event) from the test set as well as the validation on real rainfall events. We chose the heaviest rain as it could reflect the performance of our model in extreme conditions.

5.1 | Comparison of the patch aggregation methods

5.1.1 | Computational time

In Table 1, we present the average time of different patch aggregation methods of our approach. We found that a trained model significantly reduces the computational time for water depth prediction compared with that of the cellular automata-based models, using only 0.5% simulation time. For all three catchment areas, the no patch overlap option takes the least time. The computation times of using the mean value and maximum value options are close, and the time difference is less than 6.2% on average. In contrast, the median value option is the slowest because the computer needs to keep all the data in the memory before the median value can be obtained.

5.2 | Prediction accuracy

The MAEs and the error distributions of the different patch aggregation methods are presented in Figure 4. The figure shows no significant difference between the results of the different methods and suggests that choosing different aggregation methods has negligible effect on the accuracy in general. Moreover, the results of rainfall events from the test data set do not show higher MAEs than those from the training set, suggesting that our model generalizes well with rainfall variations. A more detailed analysis of the results shows that using overlapping patches generally results in lower MAEs than those with the "No overlapping" option. Using the median value usually gives better results than the Mean and Maximum value options. In addition, the histograms of the prediction error on the right-hand side of Figure 4 show that the "No overlapping" option generates more under- and over-predictions in the three catchments, suggesting that it generates a large number of outliers. As a short conclusion, using the "Mean value" option for patch aggregation shows a good balance between accuracy and prediction time.

5.3 | Prediction accuracy of the best patch aggregation method

5.3.1 | Accuracy in shallow and deep waters

The prediction accuracies of the proposed urban pluvial flood prediction approach in the three catchments are presented as 2D histograms in Figure 5. The first row shows the density plots of raster cells with simulated water depth in the x axis and predicted water depths in the y axis. The second row are density plots with simulated water depth in the x axis and prediction error in the y axis. The second row plots are essentially the 45° shear of the first row plots, but they are coloured by the ratio of cells in each simulated water depth. The histograms serve as the alternatives of scatter plots for better readability. The diagonal and horizontal dots in the plots show the baseline of an ideal model with 100% accuracy - models with higher accuracy are less divergent from the baseline. As seen, the histograms clearly show that our model produces accurate results. The first row of the figure shows a higher level of divergence in shallow-water areas than in deep-water areas. But it does not suggest lower accuracy as the plots are coloured logarithmically and the percentage of cells that have an absolute error below 0.1 m is constant (second row in Figure 5).

5.3.2 | Spatial distribution of errors

Figure 6 shows the predicted and simulated water depth of the three catchment areas, and Figure 7 shows the corresponding spatial distribution of absolute and relative errors (note that the errors are coloured non-linearly so

TABLE 1 Average time performance of the prediction model

Catchment	Catchment size (# pixel)	Pre-processing time (s)	Prediction time (s)[a]				Simulation time (s)[a]	Training time[b] (s)
			No overlapping	Mean value	Median value	Maximum value		
Luzern	3,369 × 3,110	1.898	0.678 s	2.693 s	14.749 s	2.556 s	2 h 20 min	5 h 25 min
Zurich	6,175 × 6,050	6.627	1.366 s	5.677 s	75.12 s	5.293 s	4 h 54 min	
Coimbra	1,625 × 2,603	0.636	0.242 s	0.965 s	5.048 s	0.902 s	2 h 18 min	

[a]The times are averaged and per rainfall event.

[b]For each catchment area, the amount of training data and training parameters were the same, and identical meta parameters were used; therefore, the average time is presented.

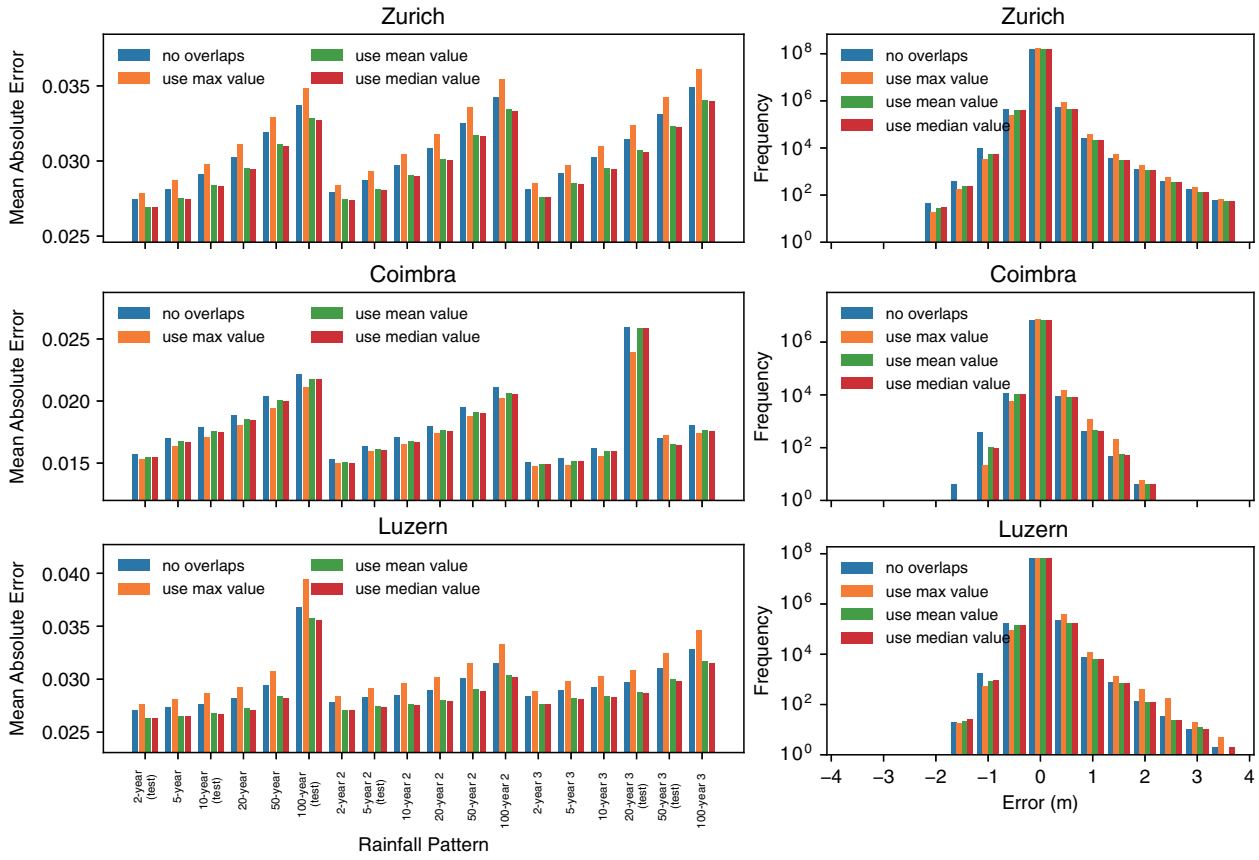

FIGURE 4 The mean absolute errors of each hyetograph (left) and the error histograms of all hyetographs (right) in the Zurich, Coimbra and Luzern catchments (top to bottom) using different aggregating methods

that the small values can be better visualized). The plots suggest that the CNNs successfully retrieved the spatial pattern of water depth, showing no significant prediction errors in the different areas. The distribution of errors is relatively even for different areas, with most raster cells an error between −0.1 m and 0.1 m. The relative errors, on the other hand, are higher for shallow-water areas than deep-water areas. Another observation seen from the enlargement areas is that the CNNs tend to smooth the output and thus are less accurate for very fine spatial scales.

The model's local performance is presented in Figure 8, in which catchments are indicated in different colors. The x marks of the figure are the maximum and minimum extents of outliers and the numbers on top are the total number of raster cells of each category. The figure shows that the performance difference is not significant. However, in a more detailed analysis, it can be seen that the model tends to perform better for flat areas (slope $< 3\%$) than steep areas (slope $\geq 15\%$), and for upstream areas (highest 33% terrain elevation) than downstream areas (lowest 33% terrain elevations). For example, in the Luzern case, raster cells that are dry in simulations are more likely to be marked as wet by the CNN prediction for downstream areas (upper right area shown in Figure 6) than upstream areas (bottom left area shown in Figure 6). The accuracy of areas around buildings is also lower than in upstream areas. The hypothesis is that the CNNs are sensitive for large input values such as the slopes caused by the radical change of elevations next to buildings and other urban features. Another observation is that the accuracy for places where there is more water than the surroundings, such as streamlines and depressions, is lower compared to other areas of the catchments. The performance difference between streamlines and depression areas is not significant.

5.3.3 | Analysis of high-error cells

It is noteworthy in Figure 5 that some raster cells that have 0 m water depth in the physically-based simulation results were over-predicted by up to 2 to 3 m. To understand what causes such high prediction error, we zoomed in to where the errors are larger than 1 m and found that these errors are due to the "smoothing" CNNs make around the radical changes of simulated water depths. These radical changes seem to be caused by the artefacts

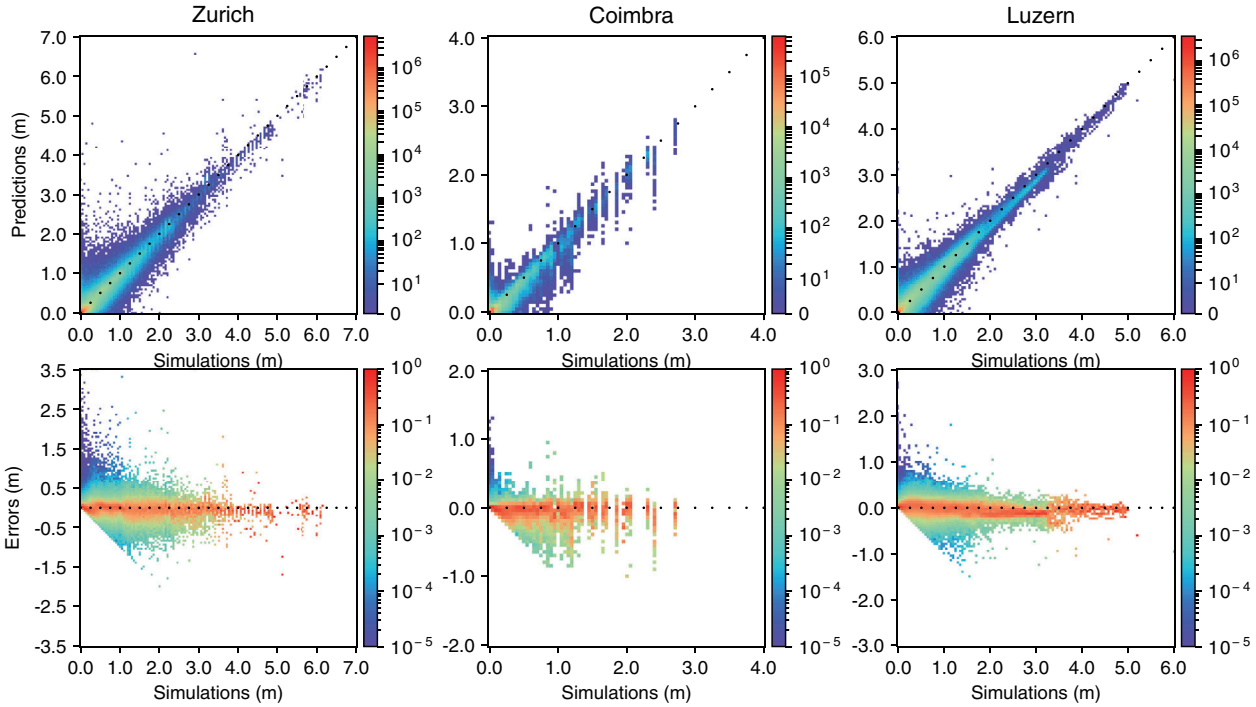

FIGURE 5 2D histograms that show the density of raster cells. The *x* axes are simulated water depths and *y* axes are predicted water depth (first row) or prediction error (second row). The first row shows the actual number and the second row shows the ratio. Note that the plots are coloured logarithmically

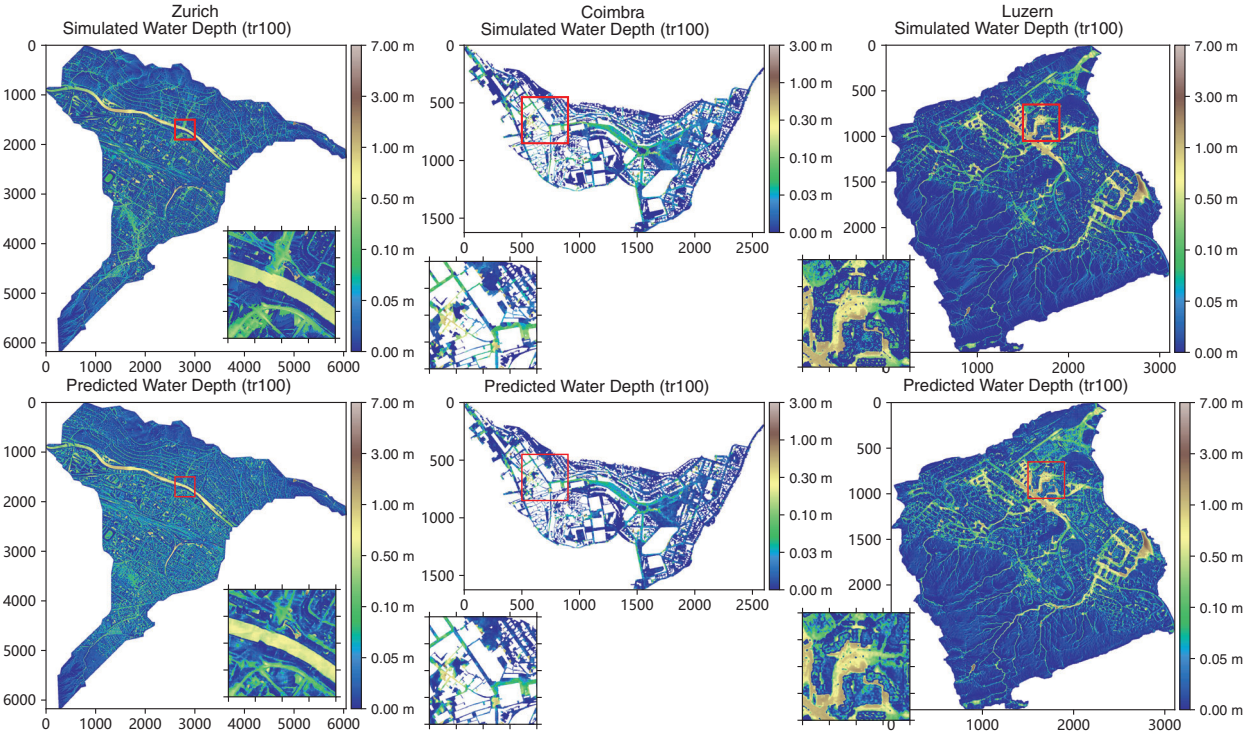

FIGURE 6 The simulated (top) and predicted (bottom) water depths of 100-year event for the Zurich, Coimbra and Luzern catchments (left to right)

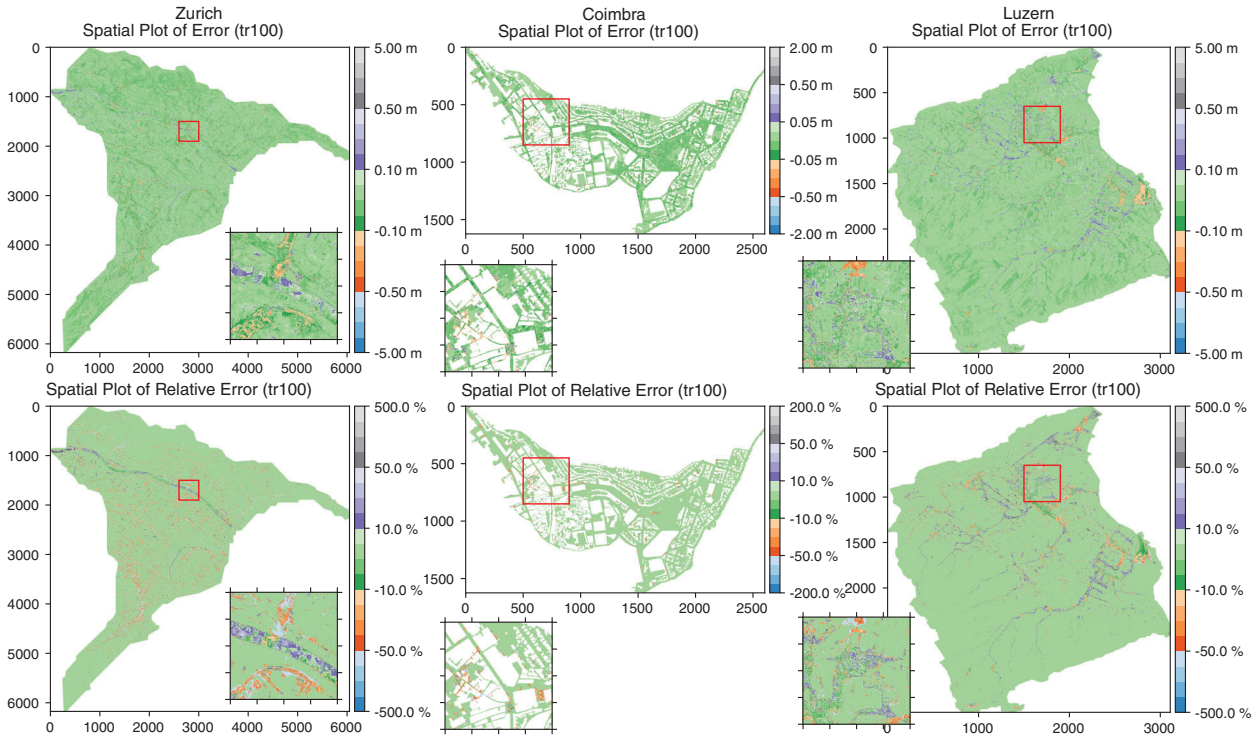

FIGURE 7 Errors (top) and relative errors (bottom) of 100-year event for all catchment areas

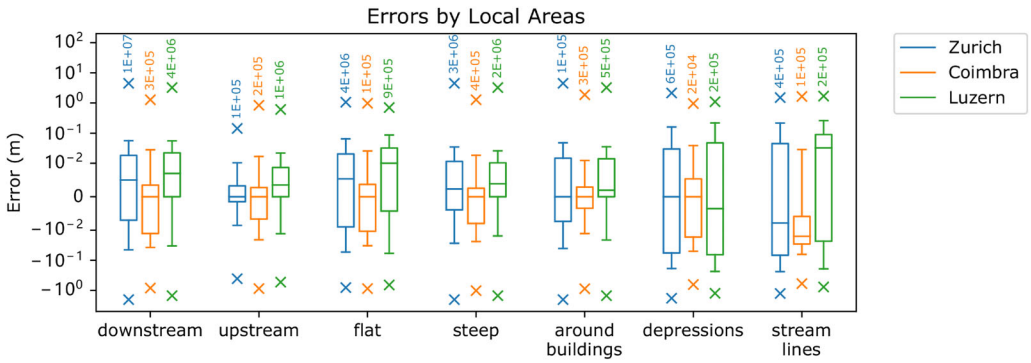

FIGURE 8 Errors by local areas for all catchment areas

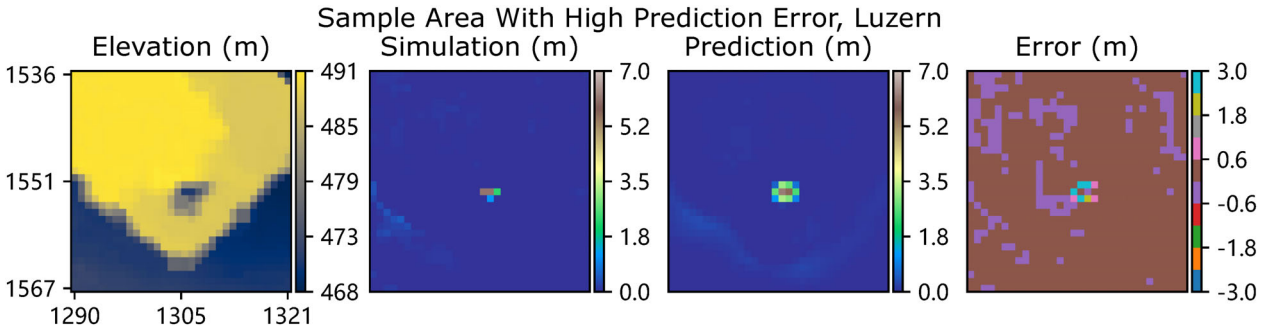

FIGURE 9 The enlargement of the area with the highest prediction error in the Luzern case

in the elevation data. An example of such cases is presented in Figure 9. Considering that each pixel has a relatively small size (1 × 1 m) compared to the size of the catchment area (more than 1,000 × 1,000 m), these errors do not affect the interpretation of the prediction result in practice as: (a) the predicted water depth in the centre is correct; (b) all high-error cells are adjacent to the centre rather than far from it; (c) a practitioner with experience can quickly identify such errors, and; (d) considering the scale and the short time, predictions containing these errors are still of high value. A detailed counting of the high-error cells and areas (group of cells with distance between each other less than 16 pixels) is shown in Table 2. These numbers presented in Table 2 can be considered small bearing in mind the millions of rasters cells of each catchment.

5.4 | Validation with real rainfall events

Figure 10 shows the validation results of our model on three real rainfall events of the Coimbra case. The results show that the predictions made by the CNN model are accurate compared to the simulations. Considering that the real rainfall events contain multiple peaks that did not exist in the training data and are thus "unfamiliar" to the CNNs, we can conclude that the model can generalize well to different hyetograph inputs and have the potential to manage arbitrary rainfall patterns.

6 | DISCUSSIONS

In this study we investigated the potential of CNNs to predict urban pluvial flooding using elevation and hyetograph data. Compared to other data-driven techniques, the main advantage of CNNs is that they can utilize spatial information and handle inputs of large areas without facing exponential growth of model parameters. Therefore, the entire catchment area can be handled by one model rather than several independent models (e.g., Berkhahn et al., 2019). Moreover, validation tests have shown that our model is accurate when provided with "unfamiliar" inputs, suggesting that the generalisability of the proposed approach is high. Additionally, unlike physically based models, which could potentially take 10 times more computational time when doubling the raster resolutions (Guidolin et al., 2016), the time increase of CNN models is expected to be linear as it only correlates with the number of input patches.

Despite the various advantages presented and discussed above, several challenges and drawbacks remain and require further investigations. The main challenge is the ability to generalize to different terrain inputs, which means a CNN model trained on one catchment area can be used in different catchment areas. Currently, this is not possible for our model, and to our knowledge, there are no other models that have achieved this goal. Another challenge is the lack of large flood datasets due to the long computational time for simulations and the difficulty of massively deploying sensors for observational data. This problem can be handled by oversampling and data augmentation techniques. Data augmentation is also beneficial for the problem of the quality of input data. For example, it is possible to address the problem of low quality or incomplete input data by adding random noise to the training data. Another limitation of the proposed approach is that the input hyetograph vectors have a fixed length, which is not suitable for rainfall events that are longer than one-hour. This problem can be solved by encoding input hyetographs using recurrent neural networks (e.g., Chang et al., 2014). Also, recent studies have suggested that recurrent neural networks can be implemented in a convolution manner to predict flood-dynamics directly (Liang & Hu, 2015). Finally, the method presented in this study can still be improved by considering other flood relevant factors such as flow velocity. This can be done by adding extra image channels to the output rasters.

TABLE 2 Raster cells with high prediction error for 100-year event in all catchments

	Zurich		Coimbra		Luzern	
Absolute errors (m)	Number of cells	Number of areas	Number of cells	Number of areas	Number of cells	Number of areas
[0.5, 1)	7,949	4,160	210	59	3,140	1,600
[1, 2)	835	354	5	4	212	101
[2, 3)	57	28	0	0	10	4
[3, ∞)	10	4	0	0	2	1

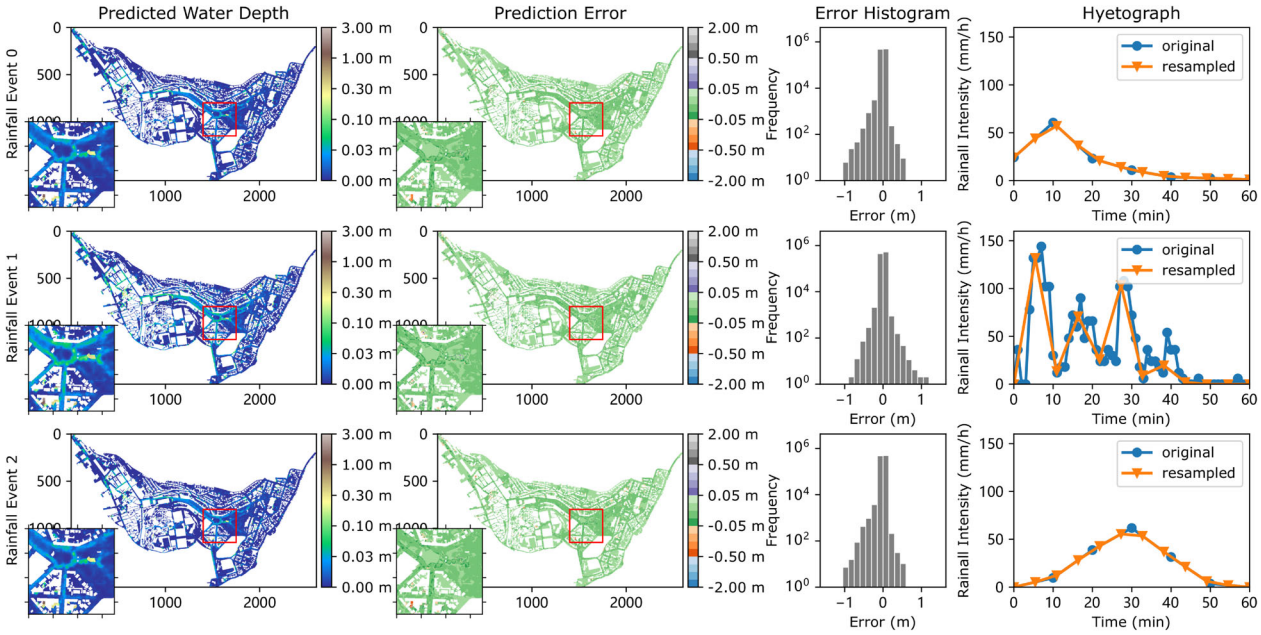

FIGURE 10 The prediction results of three real rainfall events for the Coimbra case

7 | CONCLUSIONS AND FUTURE STEPS

Computational complexity has been the bottleneck of performing systematic flood analyses using physically based models. Regarding this challenge, this paper proposes that maximum water depth predictions can be generated using CNNs as an image-to-image translation task from elevation and hyetograph inputs. The proposed approach was tested in three different catchment areas, and the results showed that the improvement in computational time was substantial and the accuracy was acceptable for practice purposes. The main contributions of this study are: (a) in contrast to other approaches, the proposed model uses spatial information to predict maximum urban pluvial flood water depth; (b) the proposed model combines both spatial and vector inputs; (c) the obtained results are accurate with the ability to generalize to different rainfall inputs, and (d) the proposed model can be potentially used to address other relevant applications without changing the network designs.

For future work, it would be interesting to extend the current investigation by combining elevation rasters of different catchments with different hyetographs, producing flood prediction models that can generalize to both terrain and rainfall inputs. Another interesting direction would be to train and validate the model using observational data to bypass the issue of long computational time for preparing the dataset. This direction is becoming possible through the combination of crowdsourcing methods (Zheng et al., 2018) and computer vision techniques (Moy de Vitry et al., 2019).

ACKNOWLEDGMENTS

This work was funded by the China Scholarship Council grant 201706090254. The authors would like to thank Águas de Coimbra for providing the drainage network information used in this study.

CONFLICT OF INTEREST

The authors declare no conflicts of interests.

DATA AVAILABILITY STATEMENT

The simulation data, source code and trained models can be obtained from the data repository (Guo, Leitão, Simões, & Moosavi, 2019) hosted by the research collection of ETH Zurich with DOI link 10.3929/ethz-b-000365484.

ORCID

Zifeng Guo https://orcid.org/0000-0002-5765-1694

ENDNOTE

[1] The rainfall events were recorded by a rain gauge owned by Aguas de Coimbra (Portugal) located within the catchment boundary.

REFERENCES

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Kudlur, M. (2016). Tensorflow: A system for large-scale machine learning. Paper presented at 12th symposium on

operating systems design and implementation. USENIX, Savannah, pp. 265–283

Bates, P. D., Horritt, M. S., & Fewtrell, T. J. (2010). A simple inertial formulation of the shallow water equations for efficient two-dimensional flood inundation modelling. *Journal of Hydrology*, *387*(1–2), 33–45. https://doi.org/10.1016/j.jhydrol.2010.03.027

Berkhahn, S., Fuchs, L., & Neuweiler, I. (2019). An ensemble neural network model for real-time prediction of urban floods. *Journal of Hydrology*, *575*, 743–754. https://doi.org/10.1016/j.jhydrol.2019.05.066

Bradbrook, K. F., Lane, S. N., Waller, S. G., & Bates, P. D. (2004). Two dimensional diffusion wave modelling of flood inundation using a simplified channel representation. *International Journal of River Basin Management*, *2*(3), 211–223. https://doi.org/10.1080/15715124.2004.9635233

Bruwier, M., Mustafa, A., Aliaga, D. G., Archambeau, P., Erpicum, S., Nishida, G., … Dewals, B. (2018). Influence of urban pattern on inundation flow in floodplains of lowland rivers. *Science of the Total Environment.*, *622-623*, 446–458. https://doi.org/10.1016/j.scitotenv.2017.11.325

Bui, D. T., Hoang, N. D., Martínez-Álvarez, F., Ngo, P. T. T., Hoa, P. V., Pham, T. D., … Costache, R. (2020). A novel deep learning neural network approach for predicting flash flood susceptibility: A case study at a high frequency tropical storm area. *Science of the Total Environment*, *701*, 134413. https://doi.org/10.1016/j.scitotenv.2019.134413

Chang, F. J., Chen, P. A., Lu, Y. R., Huang, E., & Chang, K. Y. (2014). Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *Journal of Hydrology*, *517*, 836–846. https://doi.org/10.1016/j.jhydrol.2014.06.013

Chen, A. S., Djordjevic, S., Leandro, J., & Savic, D. (2007). The urban inundation model with bidirectional flow interaction between 2D overland surface and 1D sewer networks. Paper presented at 6th international conference on sustainable techniques and strategies in urban water management (NOVATECH 2007), Lyon, France, pp. 465–472.

De Smith, M. J., Goodchild, M. F., & Longley, P. (2007). *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*, Leicester, England: . Troubador Publishing Ltd.

Feng, T., Yu, L. F., Yeung, S. K., Yin, K., & Zhou, K. (2016). Crowd-driven mid-scale layout design. *ACM Transactions on Graphics*, *35*(4), 132. https://doi.org/10.1145/2897824.2925894

Fewtrell, T. J., Bates, P. D., Horritt, M., & Hunter, N. M. (2008). Evaluating the effect of scale in flood inundation modelling in urban environments. *Hydrological Processes: An International Journal*, *22*(26), 5107–5118. https://doi.org/10.1002/hyp.7148

Gao, H., Sun, L., & Wang, J. X. (2020). PhyGeoNet: Physics-informed geometry-adaptive convolutional neural networks for solving parametric pdes on irregular domain. arXiv preprint arXiv:2004.13145

Gebrehiwot, A., Hashemi-Beni, L., Thompson, G., Kordjamshidi, P., & Langan, T. E. (2019). Deep convolutional neural network for flood extent mapping using unmanned aerial vehicles data. *Sensors*, *19*(7), 1486. https://doi.org/10.3390/s19071486

Ghimire, B., Chen, A. S., Guidolin, M., Keedwell, E. C., Djordjević, S., & Savić, D. A. (2013). Formulation of a fast 2D urban pluvial flood model using a cellular automata approach. *Journal of Hydroinformatics*, *15*(3), 676–686. https://doi.org/10.2166/hydro.2012.245

Chu, M., & Thuerey, N. (2017). Data-driven synthesis of smoke flows with CNN-based feature descriptors. *ACM Transactions on Graphics*, *36*(4), 1–14. https://doi.org/10.1145/3072959.3073643

Greydanus, S., Dzamba, M., & Yosinski, J. (2019). Hamiltonian neural networks. Paper presented at advances in neural information processing systems 32: Annual conference on neural information processing system 2019. Vancouver, Canada, pp. 15353–15363.

Guidolin, M., Chen, A. S., Ghimire, B., Keedwell, E. C., Djordjević, S., & Savić, D. A. (2016). A weighted cellular automata 2D inundation model for rapid flood analysis. *Environmental Modelling & Software*, *84*, 378–394. https://doi.org/10.1016/j.envsoft.2016.07.008

Gude, V., Corns, S., & Long, S. (2020). Flood Prediction and Uncertainty Estimation Using Deep Learning. *Water*, *12*(3), 884. https://doi.org/10.3390/w12030884

Guo, X., Li, W., & Iorio, F. (2016). Convolutional neural networks for steady flow approximation. Paper presented at the 22nd ACM SIGKDD international conference on knowledge discovery and data mining. San Francisco, pp. 481–490.

Guo Z., Leitão J. P., Simões N. E., Moosavi V. (2019). Simulation data and source code for data-driven flood emulation of urban flood. ETH Zurich Research Collection. doi: https://doi.org/10.3929/ethz-b-000365484

Hennigh, O. (2017). Lat-net: Compressing lattice Boltzmann flow simulations using deep neural networks. arXiv preprint arXiv:1705.09036.

Hochreiter, S. (1998). The vanishing gradient problem during learning recurrent neural nets and problem solutions. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, *6*(02), 107–116. https://doi.org/10.1142/S0218488598000094

Innovyze. (2019). InfoWorks ICM. Retrieved from https://www.innovyze.com/en-us/products/infoworks-icm (accessed July 30, 2019)

Jamali, B., Löwe, R., Bach, P. M., Urich, C., Arnbjerg-Nielsen, K., & Deletic, A. (2018). A rapid urban flood inundation and damage assessment model. *Journal of Hydrology*, *564*, 1085–1098. https://doi.org/10.1016/j.jhydrol.2018.07.064

Jamali, B., Bach, P. M., Cunningham, L., & Deletic, A. (2019). A Cellular Automata fast flood evaluation (CA-ffé) model. *Water Resources Research*, *55*, 4936–4953. https://doi.org/10.1029/2018WR023679

Ladický, L. U., Jeong, S., Solenthaler, B., Pollefeys, M., & Gross, M. (2015). Data-driven fluid simulations using regression forests. *ACM Transactions on Graphics*, *34*(6), 199. https://doi.org/10.1145/2816795.2818129

Kim, H. I., & Han, K. Y. (2020a). Data-Driven Approach for the Rapid Simulation of Urban Flood Prediction. *KSCE Journal of Civil Engineering*, *24*(6), 1932–1943. https://doi.org/10.1007/s12205-020-1304-7

Kim, H. I., & Han, K. Y. (2020b). Urban Flood Prediction Using Deep Neural Network with Data Augmentation. *Water*, *12*(3), 899. https://doi.org/10.3390/w12030899

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. Poster presentation in 3rd International Conference on Learning Representations (ICLR 2015), San Diego.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Leitão, J. P., Boonya-Aroonnet, S., Prodanović, D., & Maksimović, Č. (2009). The influence of digital elevation model resolution on overland flow networks for modelling urban pluvial flooding. *Water Science and Technology*, *60*(12), 3137–3149. https://doi.org/10.2166/wst.2009.754

Leitão, J. P., Zaghloul, M., & Moosavi, V. (2018). Modelling overland flow from local inflows in "almost no-time" using self-organizing maps. Paper presented at 11th international conference on urban drainage modelling (Oral presentation), Palermo, Italy.

L'homme, J., Sayers, P., Gouldby, B., Samuels, P., Wills, M., & Mulet-Marti, J. (2008). Recent development and application of a rapid flood spreading method. In: Samuels, P., Huntington, S., Allsop, W., Harrop, J. (Eds.), *Flood risk management: Research and practice*. Taylor & Francis Group, London, UK.

Liang, M., & Hu, X. (2015). Recurrent convolutional neural network for object recognition. Paper presented at the 28th IEEE conference on computer vision and pattern recognition (CVPR 2015), Boston, MA, pp. 3367–3375.

Luo, W., Li, Y., Urtasun, R., & Zemel, R. (2016). Understanding the effective receptive field in deep convolutional neural networks. In Advances in neural information processing systems 29: Annual conference on neural information processing systems 2016, Barcelona, Spain, pp. 4898–4906

Maas, A. L., Hannun, A. Y., & Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. Paper presented at the 30th international conference on machine learning (ICML 2013), Atlanta, USA, Volume 30, no. 1, p. 3.

Moy de Vitry, M., Kramer, S., Wegner, J. D., & Leitão, J. P. (2019). Scalable flood level trend monitoring with surveillance cameras using a deep convolutional neural network. *Hydrology and Earth System Sciences*, *23*(11), 4621–4634. https://doi.org/10.5194/hess-2018-570

Mustafa, A., Wei Zhang, X., Aliaga, D. G., Bruwier, M., Nishida, G., Dewals, B., … Teller, J. (2018). Procedural generation of flood-sensitive urban layouts. *Environment and Planning B: Urban Analytics and City Science*, *47*, 889–911. https://doi.org/10.1177/2399808318812458

Nair, V., & Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. Paper presented at the 27th international conference on machine learning (ICML 2010), Haifa, Israel, pp. 807–814

Ngiam, J., Khosla, A., Kim, M., Nam, J., Lee, H., & Ng, A. Y. (2011). Multimodal deep learning. Paper presented at the 28th international conference on machine learning (ICML 2011), Bellevue, pp. 689–696

Plate, E. J. (2002). Flood risk and flood management. *Journal of Hydrology*, *267*(1–2), 2–11. https://doi.org/10.1016/S0022-1694(02)00135-X

Samuels, P. G. (1990). Cross-section location in 1-D models. In W. R. White & J. Watts (Eds.), *2nd international conference on river flood hydraulics* (pp. 339–350). Chichester: Wiley.

Schuster, M., & Paliwal, K. K. (1997). Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, *45*(11), 2673–2681. https://doi.org/10.1109/78.650093

Teng, J., Jakeman, A. J., Vaze, J., Croke, B. F., Dutta, D., & Kim, S. (2017). Flood inundation modelling: A review of methods, recent advances and uncertainty analysis. *Environmental Modelling & Software*, *90*, 201–216. https://doi.org/10.1016/j.envsoft.2017.01.006

Thuerey, N., Weißenow, K., Prantl, L., & Hu, X. (2020). Deep learning methods for Reynolds-averaged Navier–Stokes simulations of airfoil flows. *AIAA Journal*, *58*(1), 25–36. https://doi.org/10.2514/1.J058291

Tompson, J., Schlachter, K., Sprechmann, P., & Perlin, K. (2017). Accelerating eulerian fluid simulation with convolutional networks. Paper presented at proceedings of the 34th international conference on machine learning (ICML 2017), Sydney, Australia, pp. 3424–3433.

Wang, Y., Fang, Z., Hong, H., & Peng, L. (2020). Flood susceptibility mapping using convolutional neural network frameworks. *Journal of Hydrology*, *582*, 124482. https://doi.org/10.1016/j.jhydrol.2019.124482

Zaghloul, M. (2017). Machine-learning aided architectural design - synthesize fast CFD by machine-learning. PhD dissertation. ETH Zurich. https://doi.org/10.3929/ethz-b-000207226

Zheng, F., Thibaud, E., Leonard, M., & Westra, S. (2015). Assessing the performance of the independence method in modeling spatial extreme rainfall. *Water Resources Research*, *51*(9), 7744–7758. https://doi.org/10.1002/2015WR016893

Zheng, F., Tao, R., Maier, H. R., See, L., Savic, D., Zhang, T., … Popescu, I. (2018). Crowdsourcing Methods for Data Collection in Geophysics: State of the Art, Issues, and Future Directions. *Reviews of Geophysics*, *56*(4), 698–740. https://doi.org/10.1029/2018RG000616

How to cite this article: 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 Management*. 2021;14:e12684. https://doi.org/10.1111/jfr3.12684