

Article

A Convolutional Neural Network for Steady-State Flow Approximation Trained on a Small Sample Size

Guodong Zhong^{1,2}, Xuesong Xu³, Jintao Feng³ and Lei Yuan^{1,2,*}¹ School of Architecture and Urban Planning, Shenzhen University, Shenzhen 518060, China² Shenzhen Key Laboratory for Optimizing Design of Built Environment, Shenzhen University, Shenzhen 518060, China³ The Institute of Architecture Design & Research, Shenzhen University, Shenzhen 518060, China

* Correspondence: yuanlei@szu.edu.cn

Abstract: The wind microclimate plays an important role in architectural design, and computational fluid dynamics is a method commonly used for analyzing the issue. However, due to its high technical difficulty and time-consuming nature, it limits the interaction and exploration between designers and environment performance analyses. To address the issue, scholars have proposed a series of approximation models based on machine learning that have partially improved computational efficiency. However, these methods face challenges in terms of balancing applicability, prediction accuracy, and sample size. In this paper, we propose a method based on the classic Vggnet deep convolutional neural network as the backbone to construct an approximate model for predicting steady-state flow fields in urban areas. The method is trained on a small amount of sample data and can be extended to calculate the wind environment performance. Furthermore, we investigated the differences between geometric representation methods, such as the Boolean network representation and signed distance function, as well as different structure models, such as Vgg-CFD-11, Vgg-CFD-13, Vgg-CFD-16, and Vgg-CFD-19. The results indicate that the model can be trained using a small amount of sample data, and all models generally possess the ability to predict the wind environment. The best performance on the validation set and test set was achieved with an RMSE (Root Mean Square Error) of 0.7966 m/s and 2.2345 m/s, respectively, and an R-Squared score of 0.9776 and 0.8455. Finally, we embedded the best-performing model into an architect-friendly urban comprehensive analysis platform, URBAN NEURAL-CFD.

Keywords: deep learning; performance simulation; computational fluid dynamics; convolutional neural networks; surrogate modeling; small sample size



Citation: Zhong, G.; Xu, X.; Feng, J.; Yuan, L. A Convolutional Neural Network for Steady-State Flow Approximation Trained on a Small Sample Size. *Atmosphere* **2023**, *14*, 1462. <https://doi.org/10.3390/atmos14091462>

Academic Editor: Massimiliano Burlando

Received: 21 July 2023

Revised: 21 August 2023

Accepted: 13 September 2023

Published: 20 September 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

1.1. Building the Wind Environment

High-density cities face problems such as urban heat islands and air pollution, and natural ventilation can play a key role in addressing such issues. To analyze the relationship between a design plan and airflow field, computational fluid dynamics (CFD) methods are typically used. In the past few decades, significant progress has been made in simulating flow problems using the numerical discretization of the Navier–Stokes equations. Compared with wind tunnel experiments and field measurements, it has advantages in terms of low cost and high accuracy [1] and has been widely used. However, this cost advantage is only relative and the time required to obtain results still extends to hours or days [2]. However, different from the working logic of designers, CFD calculations require tedious setting of physical parameters, so existing airflow simulation tools are rarely integrated intuitively into the tools designers commonly use [3].

To address this issue, scholars have proposed more efficient algorithms [4]. For instance, fast fluid dynamics is a technique for solving incompressible Navier–Stokes

equations and has been applied in fluid visualization for video games [5]. In addition, when applied to GPUs (Graphics Processing Units), it can achieve real-time information for small buildings, which can be 500–1500 times faster than traditional CFD methods based on CPUs [6]. Some techniques originally applied in the industrial and engineering fields, such as the porous media model, have also been used to simulate the urban wind environment [7]. These techniques involve simplifying the geometry of cities and dividing street canyons into interconnected regions, helping to evaluate the ventilation and pollutant dispersion between blocks [4]. Particle-based fluid dynamic methods have also been proposed, such as Tao's use of the boundary-lattice Boltzmann method to study the impact of particles on the wall boundary layer, which is based on the collision and propagation of particles and numerical simulation of a discrete Boltzmann equation [8]. These algorithms solve the time–cost problem to a large extent and even provide immediate feedback, but it is usually difficult to obtain reliable calculations in building design scenarios using these models and further research and development is needed [9].

1.2. Development Combined with Machine Learning

For planners or architects, calculating the wind environment comfort during the design process often revolves around simple geometric shapes as the main elements of the wind field. Compared to other applications of fluid mechanics, the calculation scenarios are relatively simple, enabling further simplification of the process [10]. Machine learning provides a powerful information-processing framework that can augment current lines of fluid mechanics research [11]; scholars have begun to explore the use of machine learning methods to solve related problems. For example, for natural ventilation in high-rise residential buildings, Gan proposed a data-driven physically based model that predicts air change per hour in various areas through the surface wind pressure of the high-rise building [12]. Similarly, Zhou and Ooka used a deep neural network to predict the indoor airflow distribution based on the inlet velocity, temperature, and window surface temperature [13]. Similar to statistical methods, these methods attempt to describe case variations through a small number of parameters to seek a functional relationship between parameters and the target variable. Therefore, strong scenario restrictions exist, making it challenging to apply these methods on changing spatial forms in design.

The emerging convolutional neural networks and generative adversarial networks in the field of computer vision provide some solutions. For example, Musil proposed the ResNet for approximating real-time prediction of three-dimensional steady-state conditions to predict the wind speed distribution around building structures within a wind field range of 256 m × 128 m × 64 m, under specific wind speed conditions [14]. Mokhtar trained a conditional generative adversarial network approach using 2800 cases as a surrogate model, which can predict the pedestrian wind environment of different building forms in seconds [9]. Duering also employed a similar method and further encapsulated the model into an easy-to-use simulation tool that has been applied in performance-driven optimization design [15]. These methods treat the building morphology and its wind speed distribution as image objects, using a U-net convolutional neural network or a generative adversarial network for image-to-image model construction for an alternative model. The aforementioned research has demonstrated the effectiveness of these methods in scenarios involving early-stage variations in building forms. However, it does have two drawbacks. Firstly, in this computational scenario, the wind field range needs to match the input dimensions of the neural network model, rather than being adjusted based on the size of the building environment. Typically, a large training dataset composed of structured data is required, including urban morphologies and environmental performance distributions, resulting in high computational costs. Karniadakis proposed a physics-informed neural network approach that couples data-driven methods with various Navier–Stokes formulations used in fluid calculations [16,17]. It allows the neural network to be trained using residual values that adhere to physical laws, such as the conservation laws, subject to kinematic or thermodynamic constraints [18]). Although this method generally

achieves high accuracy, the neural network model needs to be built upon partial differential equations and precomputed data of physical fields. Therefore, when there are changes in the physical field, the neural network also needs to be retrained, and currently it is rarely seen in applications within the field of architectural wind environments.

Table 1 lists the typical studies of the aforementioned. These studies essentially dedicated to achieving rapid predictions of environmental fluid dynamics within a specific scenario, sacrificing accuracy to avoid costly CFD simulations. It is challenging to simultaneously achieve optimal conditions for the model scenario diversity, prediction accuracy, and required sample size. However, we can strive to achieve a better balance among these three factors, integrating the substitute model more effectively into the workflow of architects.

Table 1. Typical studies in the Field.

Type of Method	Reference	Diversity of Applicable Scenarios	Accuracy of Prediction Problems (on Test Dataset)	Sample Size
Artificial neural network	[19]	investigating the relationship between CO ₂ concentration and environmental parameters	79.3%	2760
	[12]	investigating the relationship between plan shapes, surface pressure distribution, and the air change per hour.	MAE = 21.3% MAPE = 43.1%	600
	[13,20]	investigating the relationship between inlet vent speed and distribution of velocities within a specific room	R2 = 0.97	Sampling from 5 cases
Convolutional neural network	[14]		no quantitative expression	3325
	[21]		no quantitative expression	8800
	[2]	investigating the distribution of wind speeds within a specific wind field	Relative error = 1.76%	100,000
Conditional generative adversarial networks	[10]		MAE = 0.3 m/s (initial wind speed = 6 m/s)	15,000
	[22]		R2 = 0.70	9290 (data augment from 1858 cases)
Physics-informed neural networks	[17]	the multi-physics with initial and boundary conditions known	no quantitative expression	6000

1.3. Objectives and Structure

Machine learning methods, as an alternative approach mentioned earlier, have been proven to be a feasible technique. These methods learn the relationship between the input feature vector representing the geometry shape and the corresponding wind field to obtain an approximate model for wind prediction. However, these methods are difficult to balance in terms of the diversity of applicable scenarios, the accuracy of prediction problems, and the required sample size.

This paper focuses on the calculation of steady laminar flow in urban streets and aims to achieve the following objectives: (1) Improve data utilization by modifying the model's data handling method and develop a predictive model that requires a small sample size. (2) Investigate the impact of neural network structures and data input methods on

the performance of the approximation model. (3) Develop a fast CFD prediction tool for architects, using the neural network model as its core.

This article proposes a widely applicable convolutional neural network model for predicting street-level wind environments, which adopts a sampling approach centered around measurement points allowing for training alternative models with a small number of training cases. The model takes different convolutional neural network structures and spatial representation methods are compared to understand their impact. The best model is selected and encapsulated into a Grasshopper plugin named URBAN NEURAL, which enables fast calculation of street-level wind environment profiles. As shown in Figure 1, the steps required are Step 1: data collection, Step 2: neural network modeling and training, Step 3: post-processing, Step 4: model performance, and Step 5: encapsulation. In this paper, the Methodology section focuses on describing Steps 1–3, while the Result and Analysis section focuses on Steps 4 and 5.

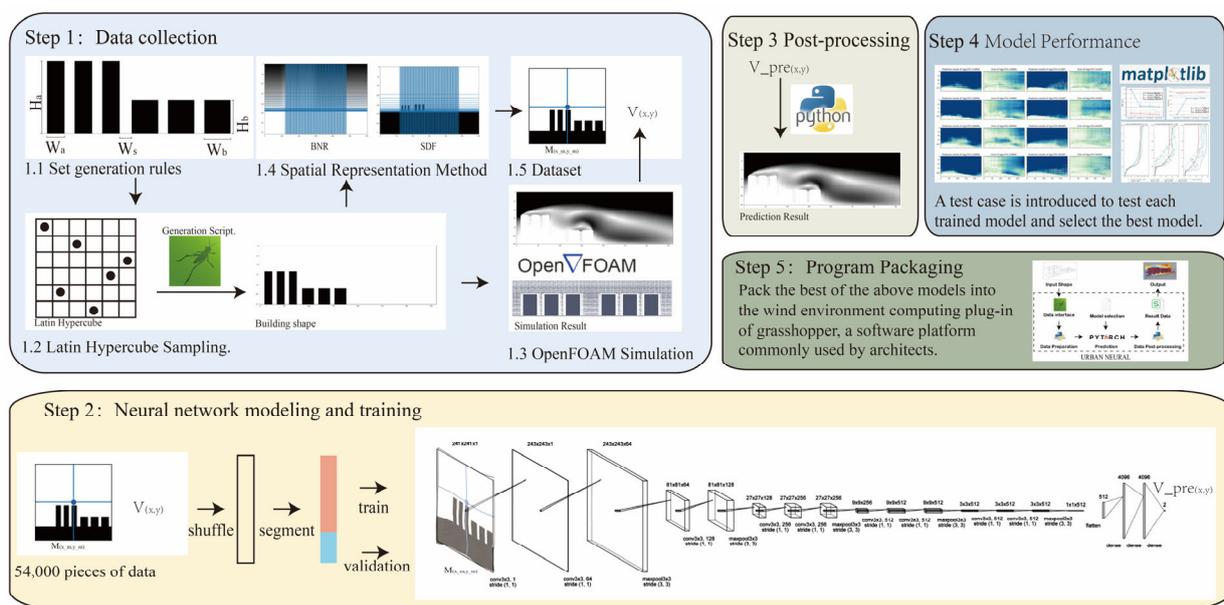


Figure 1. Workflow.

2. Methodology

In this paper, we propose a fluid dynamics approximation model, Vgg-CFD, constructed based on the VGG (Visual Geometry Group) model [23], a convolutional neural network (CNN) model that has achieved success in the field of image vision. In contrast to other CNN architectures such as ResNet and Inception, the VGG model offers a more coherent and succinct framework, facilitating our exploration of the role played by convolutional layers in the process of improving neural networks. Convolutional layers can significantly improve the model’s analytical ability for spatial features and have strong generalization ability, which may be related to the characteristics of convolutional layer parameters such as “local connectivity, parameter sharing” [24]. Some studies have proven its effectiveness. For example, Guo used a CNN method to construct an approximate model of the flow field of a car’s cross-section [2] and Bhatnagar proposed an approximate flow field model for studying wing shapes [25]. However, these methods require thousands of flow field cases and data to achieve results, greatly increasing the workload. In our previous research [26], we found that the neural network model established in traditional research methods is based on the relationship between spatial fields and performance fields and the data are not fully used. When a model is established from the perspective of points, the contact between the surrounding space of a point and the value of that point can increase the training set scale and it is possible to use the dataset more effectively. The theoretical premise of this

method is that, under a certain wind field, the spatial form within a certain range of the point is the determining factor of the point’s wind speed vector value.

To more intuitively demonstrate the function relationship based on the views of points, a random 2D steady-state wind field calculation case was used. The wind field approached from the left and the relevant settings are the same as those in Section 2.1.3. Figure 2a shows the wind speed map of the wind field. We uniformly sampled the space of the wind field at intervals of 1 m, and Figure 2b shows the scatter plot of the horizontal and vertical wind speeds at each measurement point. Furthermore, we selected the part of the wind speed distribution that is more concentrated, which is the range of the horizontal wind speed (−5 m/s, 15 m/s) and the vertical wind speed (−2 m/s, 2 m/s), and constructed a matrix according to the wind speed magnitude. Each subgraph in the matrix characterized the spatial pattern within 60 m around the point, as shown in Figure 3. In the subgraphs, the blue dots represent the measurement points and the black area represents the building or the ground.

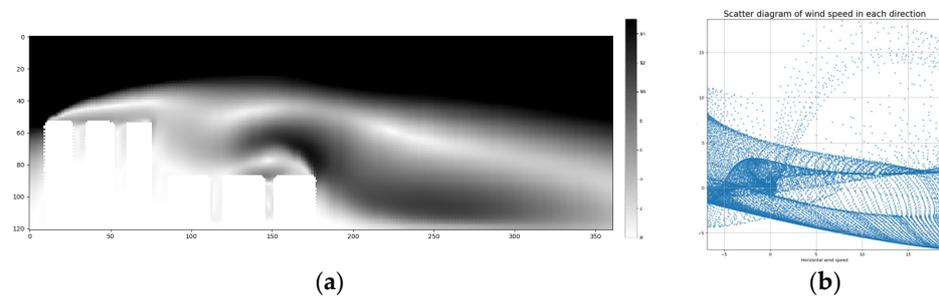


Figure 2. (a) Velocity map and (b) scatter diagram of wind speed in each direction.

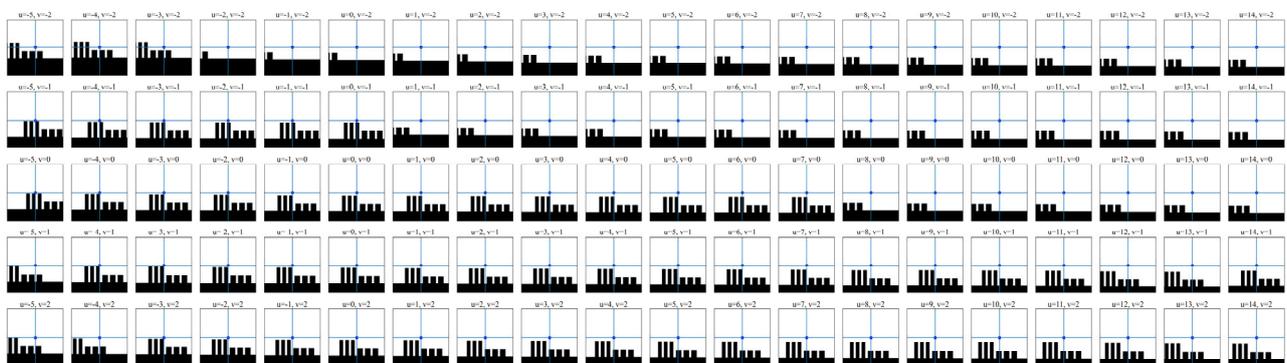


Figure 3. Spatial matrix arranged according to the wind speed component magnitude.

It can be observed from the wind speed component matrix that the proportion of white color increases as the subgraphs move toward the upper-right corner of the matrix and the position of the measurement point becomes closer to the obstacle in the lower subgraphs. However, this relationship is not clear and is difficult to describe. It is reasonable to speculate that under a constant wind field, there exists a higher-dimensional, nonlinear relationship between the spatial configuration of the surrounding environment of a measurement point and its corresponding wind speed value. The underlying relationship can be expressed through Formulas (1) and (2). For any point (x, y) in space, the corresponding surrounding environment $S_{(x,y)}$ can be represented using a 2D matrix formed using (x_m, y_m) in the surrounding area and with (x_n, y_n) as the precision. Here, x_m, y_m, x_n, y_n are natural numbers greater than 0. As x_m, y_m approach infinity and x_n, y_n approach 0, $Sur_{(x,y)}$ can fully reflect the surrounding environment of the point. However, due to computational cost, in this paper, (x_m, y_m) is set to $(120, 120)$ and (x_n, y_n) is set to $(1, 1)$, indicating that for any point in space, a g function is used to characterize the surrounding environment with a minimum cutting unit of $(1,1)$ and a surrounding range of $(120,120)$.

Sections 2.1.4 and 2.1.5 describe the g function, as shown in Formula (1). Additionally, under a constant incident wind field v_0 , the wind speed value $V_{(x,y)}$ corresponding to any point (x,y) in space is only related to the surrounding environment $S_{(x,y)}$ and its function can be represented as f_{v_0} , as shown in Formula (2). In this paper, v_0 is set to the incident wind speed with a reference surface wind speed value of 6 m/s at a height of 10 m. For details of the setting, see Section 2.1.3.

$$Sur_{(x,y)} \approx g\left((x,y), M_{(x_m,y_m)}, S_{(x_n,y_n)}\right) \tag{1}$$

$$V_{(x,y)} = f_{v_0}\left(Sur_{(x,y)}\right) \tag{2}$$

In the following section, we will concretize this process through data preparation, model construction and training, and data post-processing.

2.1. Data Collection

2.1.1. Set Generation Rules

In this paper, we chose to discuss case studies focusing on street canyon profiles with simpler wind field characteristics due to the following advantages: (1) Street canyon profiles have a simpler and more easily generalized wind environment morphology compared to complex and diverse urban building forms, which lack clear morphological generation logic. (2) Street canyon profiles exhibit distinct characteristics in terms of wake zones, turbulent zones, and separation zones, making it advantageous for observing the fitting capability of the proposed alternative algorithms in different flow field features. (3) Calculating the wind environment in street canyon profiles is relatively straightforward, and the data required for an analysis and description are readily available.

Urban blocks in cities vary greatly in shape, and we simplified the block form in a reasonable way. As shown in Figure 4, the hypothetical block consists of two groups of blocks and their constituent streets, with three buildings in each of the two groups of blocks, and the spacing between any two buildings in a block is 20 m. The blocks are controlled by five morphological parameters: the height of building A (H_a), the width of building A (W_a), the height of building B (H_b), the width of building B (W_b), and the width of the block (W_s). Taking into account the typical situation of Chinese urban streets, we defined the threshold values for the five parameters, as shown in Table 2. These five parameters were used to generate building profile models on the Grasshopper platform.

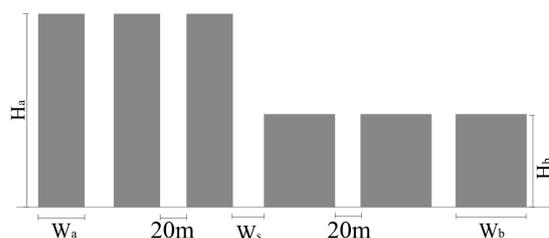


Figure 4. Illustration of generated street block forms.

Table 2. Range of variation for each variable.

	Building A		Building B		Street Valley
	Height	Width	Height	Width	Width
Range (m)	[9,80]	[6,30]	[9,80]	[6,30]	[7,45]

2.1.2. Latin Hypercube Sampling

In this section, we describe how we used the aforementioned parametric generation platform to sample and generate street form cases. The spatial distribution of the training

set samples helped determine the generalization ability of the neural network model to some extent [27]. To better cover the potential sample space, we used the Latin hypercube sampling (LHS) method. Compared to the random sampling method, the LHS method can generate a more representative sample space and increase the multidimensional uniformity [28]. For the setup of the LHS method, we referred to Shen and Han [29]. A total of 25 sets of sample parameters were obtained and are presented in Table 3. These parameters were used to generate the cases used in the Grasshopper platform through the morphogenetic generator in Section 2.1.1.

Table 3. The parameters of the training set sample cases.

No.	Building A (m)		Building B (m)		Street Valley (m)
	Height	Width	Height	Width	Width
Range	[9,80]	[6,30]	[9,80]	[6,30]	[7,45]
1	66	15	31	23	12
2	39	9	44	13	33
3	31	28	56	9	36
4	48	24	28	24	14
5	25	23	39	6	17
6	52	6	75	15	36
7	10	21	63	19	30
8	74	20	68	10	18
9	23	17	59	27	44
10	42	18	72	12	29
11	51	27	14	29	32
12	44	22	22	28	9
13	65	12	43	14	12
14	75	29	19	18	20
15	32	10	60	11	28
16	29	9	46	9	16
17	36	19	52	23	21
18	69	7	15	26	39
19	12	25	77	17	43
20	60	14	50	19	38
21	61	12	35	8	40
22	15	13	11	22	24
23	79	29	33	27	27
24	19	16	71	21	8
25	56	25	25	16	23

2.1.3. CFD Simulation

CFD calculations were used to obtain the wind field information of the samples. To save computational time, only the cross-section of the street space as a 2D wind field was considered. All cases were simulated using the OpenFOAM software (Version 5.0, available at www.openfoam.com, released by OpenFOAM Foundation) for steady-state RANs equations, adopting the k- ϵ turbulence model. Grid partition and wind field settings were determined in line with the related literature and industry standards [30,31]. The

distance between the windward and top surfaces of the building and the building's distance from the leeward surface were defined as 5 times and 15 times the height of the highest building, respectively. The lateral boundaries were set as symmetrical. The inlet wind speed was set using the gradient wind, with a speed of 6 m/s at a reference height of 10 m, assuming a flat terrain and using the empirical urban environment landscape roughness. The roughness settings can be found in Hammond et al. [32] and Wieringa [33]. The grid partition was carried out using SnappyHexMesh, with a final grid count of approximately 4.5 million cells. Figure 5 displays the computational fluid dynamics grid used for the example case. The convergence residual was set to 10^{-6} and the maximum number of iterations was set to 2000. The project was run on a computer equipped with an AMD Ryzen 7 3700X 8-Core processor (Santa Clara, CA, USA) for 130 h. The wind field information we obtained corresponding to the 25 cases is shown in Figure 6.

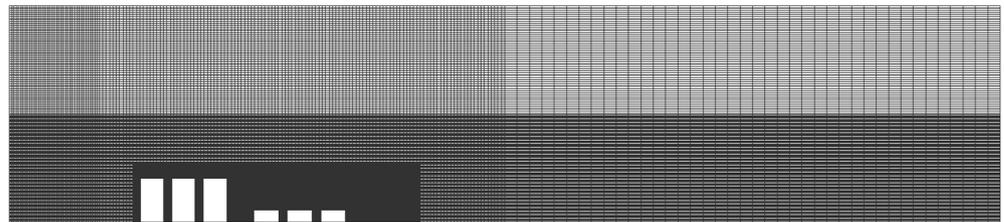


Figure 5. CFD calculation grid for a case.

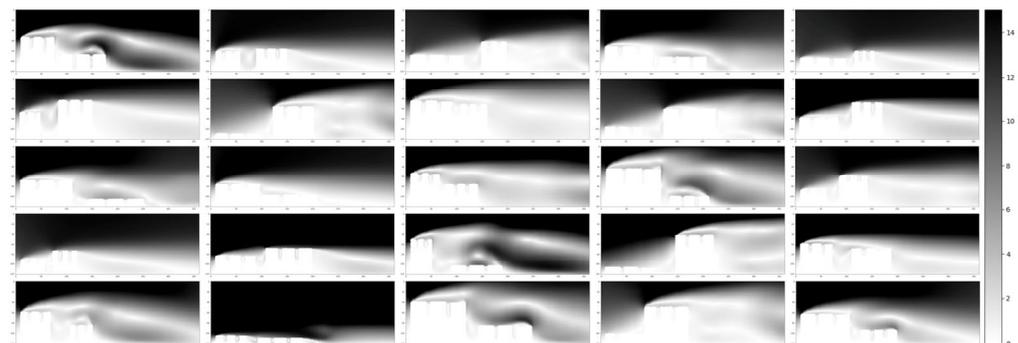


Figure 6. Wind speed maps for the case set.

2.1.4. Spatial Representation Method

The geometric model needed to be translated to obtain data recognizable using the neural network. In this section, two translation methods are presented that enable the aforementioned case model to form matrix data the model can recognize. There are various representation methods for geometric models, such as NURBS(Non-Uniform Rational B-Splines Modeling), point-based models, and triangle meshes [34]. However, these methods have large data sizes and complex structures, which greatly increase the complexity of the neural network. In this paper, we propose to use two methods, the Boolean network representation or BNR [34] and signed distance function or SDF [2], both of which are built in the same Cartesian coordinate system, as shown in Figures 7 and 8.

BNR: In a Cartesian coordinate system with an accuracy of 1 m, the map covers a horizontal range of 360 m and a vertical range of 120 m. The grid point with a value of 1 represents the obstacle, while the grid point with a value of 0 represents no obstacle, forming a 2D matrix with dimensions of (120, 360), as shown in Figure 7.

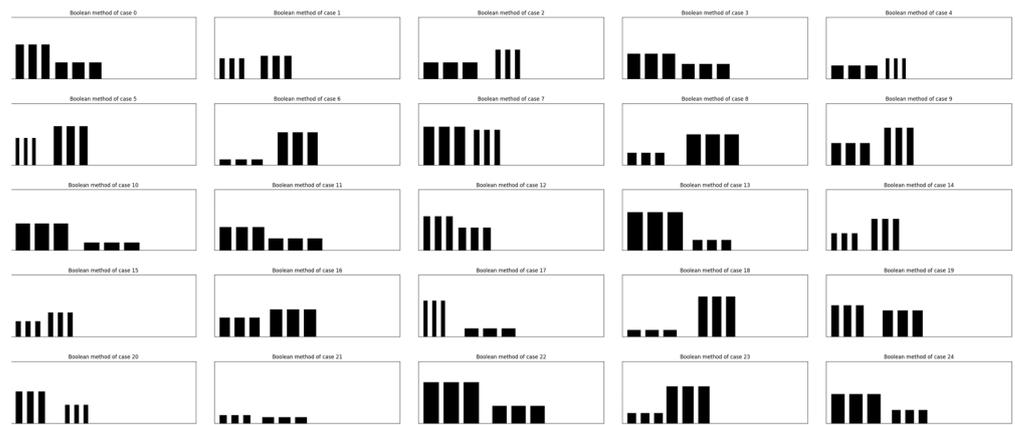


Figure 7. Boolean method expression of 25 cases in the training set.

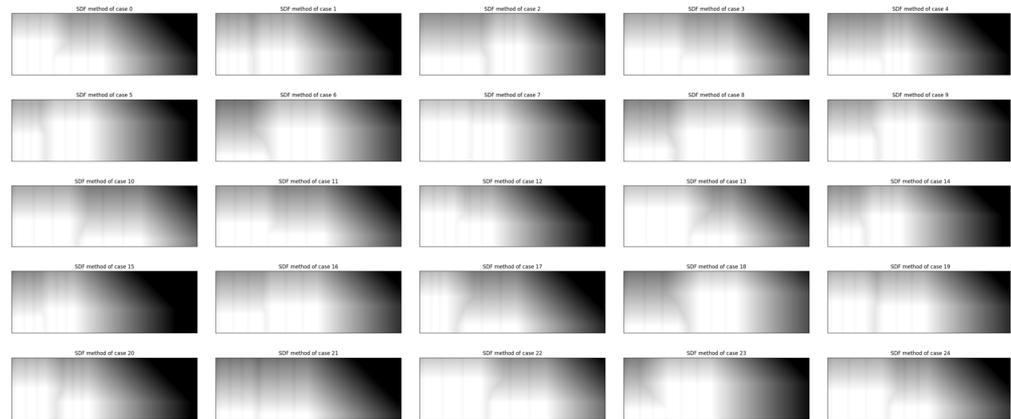


Figure 8. SDF expression of 25 cases in the training set.

SDF: In a Cartesian coordinate system with an accuracy of 1 m, the map covers a horizontal range of 360 m and a vertical range of 120 m. The grid point where the obstacle is located has a value of 0, and the value of the grid point with no obstacle is equal to the minimum distance between that point and the nearest obstacle. The minimum distance is determined with the total number of grid units moved horizontally or vertically from that point, as shown in Figure 8.

These two methods were automated using Python scripts to encode them separately. The visualization outcomes are shown in Figures 7 and 8.

2.1.5. Dataset

In wind-environment-related studies, greater attention is paid to the near-surface wind speed. Therefore, non-uniform grids are usually used in the grid division process to locally refine the target area and allocate more computational resources to the region of greater interest. Similarly, the distribution of sampled data will directly affect the bias of the approximation model’s data processing capabilities. Thus, in this study, we chose non-uniform grids for sampling. Firstly, we padded the building form data by extending them by 120 m in all directions of the section, the goal being to ensure that the sampling points on the edges of the section can still capture the surrounding environmental data. Then, we sampled at intervals of 2 m in the horizontal direction and at values rounded to the nearest integer of 1.2 to the power of a multiple in the vertical direction. Figure 9 displays the resulting sampling points, with the blue lines indicating the sampling points of the project. A total of $21 \times 180 = 2160$ data points were sampled for each case. Each data point consisted of two parts, with a data structure of $(Sur_{(x,y)}, V_{(x,y)})$, where $Sur_{(x,y)}$ had a dimension of (121,121) and $V_{(x,y)}$ had a dimension of (2).

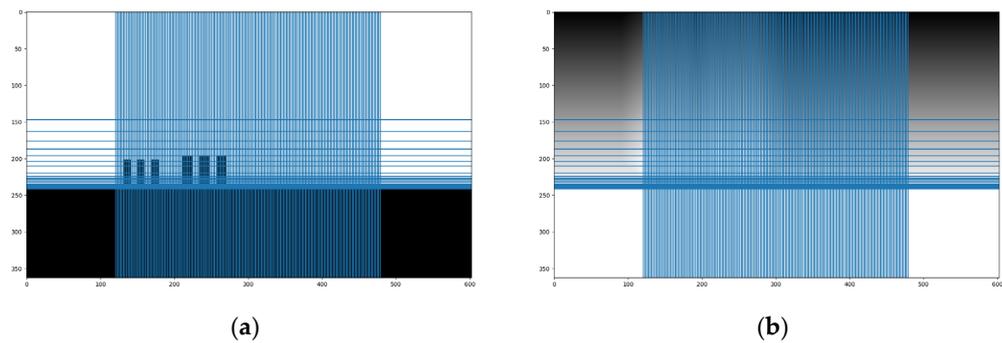


Figure 9. Sampling of the training case dataset: (a) sampling slices of the BNR representation cases; (b) sampling slices of the SDF representation cases.

After the above steps, we operated on the 25 cases shown in Figures 7 and 8 using the two sampling methods illustrated in Figure 9, resulting in a total of 54,000 samples for the dataset.

And after we removed the points inside the buildings, we obtained 51,447 samples. These samples were shuffled randomly and divided into two parts, 70% (36,013 samples) for the training set and 30% (15,434 samples) for the validation set, to be used in model training.

2.2. Model Construction and Training

The Vgg-CFD model proposed in this paper is based on the Vgg model (Simonyan and Zisserman, 2015), which is a deep convolutional neural network with a simple and stackable design. Necessary modifications were made as shown in Table 4. To investigate the influence of the depth of convolutional neural networks on their spatial performance prediction ability, we constructed four sets of models, Vgg-CFD-11, Vgg-CFD-13, Vgg-CFD-16, and Vgg-CFD-19, based on the Vgg model. Shown in Figure 10, all of them consist of 5 blocks and 4 fully connected layers, and each block consists of convolutional layers, a pooling layer, a normalization layer, and an activation function. This structure is designed to enhance the model’s nonlinearity and stability and thus accelerate the model training process. Figure 11 shows the data transmission diagram of Vgg-CFD-11, where the input parameter is $Sur_{(x,y)}$, with a dimension of (241,241), representing the surrounding environmental area with a square of 241 m centered on the measurement point, and the output parameter is $V_{(x,y)}$, with a dimension of (2), representing the corresponding wind speed values of the horizontal and vertical wind speeds at that point.

Table 4. Vgg-CFD convolutional neural network configuration.

Vgg-CFD-11	Vgg-CFD-13	Vgg-CFD-16	Vgg-CFD-19
11 Weight Layers	13 Weight Layers	16 Weight Layers	19 Weight Layers
Input matrix = (241,241)			
Conv3-64	Conv3-64 Conv3-64	Conv3-64 Conv3-64	Conv3-64 Conv3-64
Maxpool, kernel_size = 3, stride = 3			
Conv3-128	Conv3-128 Conv3-128	Conv3-128 Conv3-128	Conv3-128 Conv3-128
Maxpool, kernel_size = 3, stride = 3			
Conv3-256 Conv3-256	Conv3-256 Conv3-256	Conv3-256 Conv3-256 Conv3-256	Conv3-256 Conv3-256 Conv3-256 Conv3-256

Table 4. Cont.

Vgg-CFD-11	Vgg-CFD-13	Vgg-CFD-16	Vgg-CFD-19
11 Weight Layers	13 Weight Layers	16 Weight Layers	19 Weight Layers
Maxpool, kernel_size = 3, stride = 3			
Conv3-512	Conv3-512	Conv3-512	Conv3-512
Conv3-512	Conv3-512	Conv3-512	Conv3-512
Maxpool, kernel_size = 3, stride = 3			
Conv3-512	Conv3-512	Conv3-512	Conv3-512
Conv3-512	Conv3-512	Conv3-512	Conv3-512
Maxpool, kernel_size = 3, stride = 3			
Fully connected layer = (512, 4096)			
Fully connected layer = (4096, 4096)			
Fully connected layer = (4096, 2)			
Output matrix = (2)			

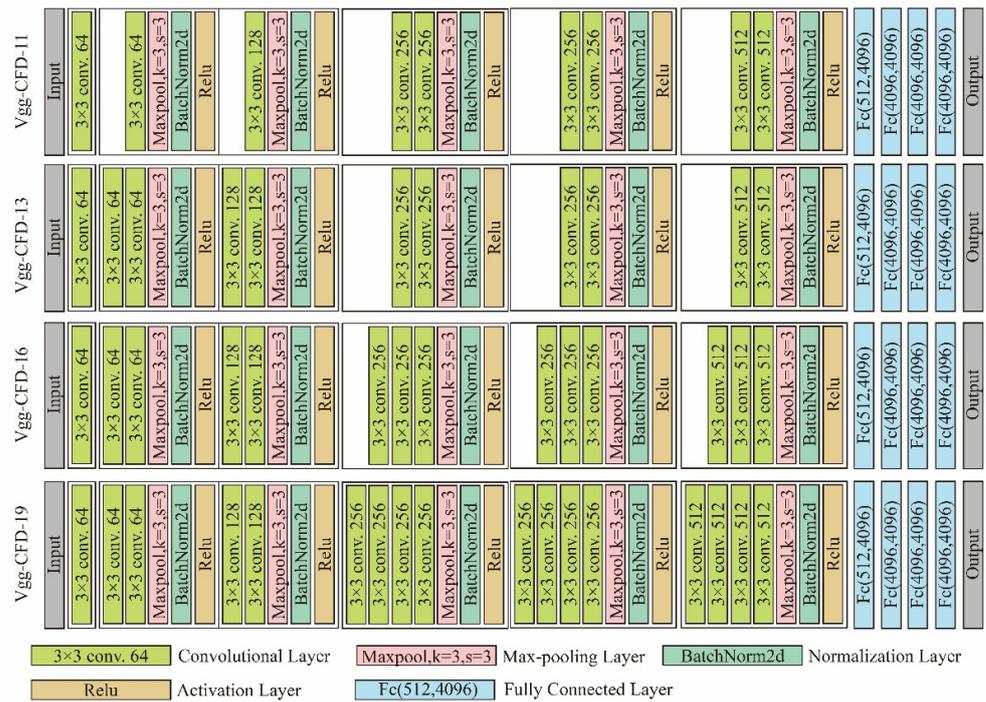


Figure 10. Vgg-CFD model structure diagram.

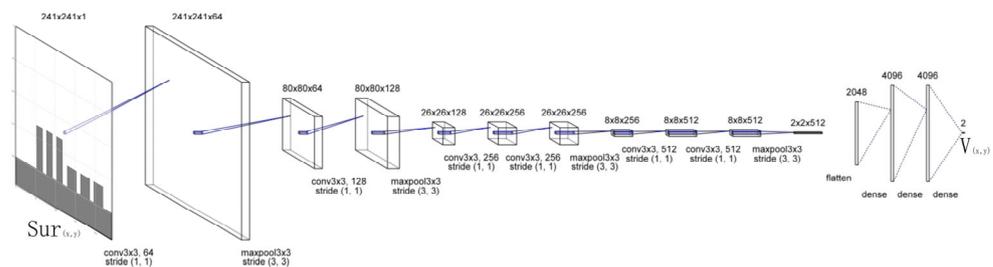


Figure 11. The data flow of Vgg-CFD-11.

The loss function for all models was set as the mean square error (MSE), which is a commonly used metric for measuring prediction errors. Additionally, all models were trained using the ADAM optimization algorithm [35]. The learning rate used to update the model weights was 10^{-5} , and the maximum number of iterations was set to 100 epochs. Finally, the models were trained on a Windows 10 computer equipped with an NVIDIA GeForce RTX 2600 SUPER graphics card (Santa Clara, CA, USA).

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2 \tag{3}$$

2.3. Post-Processing of Data

After the wind speed vector data $V_{pre(x,y)}$ of each measuring point were obtained as described above, they were rearranged on the basis of the original coordinates (x, y) to obtain the wind field vector data. These data were plotted as wind speed contour maps using Matplotlib or visualized using the Grasshopper platform through Python scripts.

3. Result and Analysis

3.1. Performance on Training and Validation Set

In this section, we will evaluate and analyze the performance of the proposed model using two metrics, namely the Root Mean Squared Error (RMSE) and R^2 score (R-Squared score), as shown in Formulas (4) and (5), respectively.

$$RMSE = \sqrt{\frac{1}{m} \sum_{i=1}^m (y_i - \hat{y}_i)^2} \tag{4}$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \tag{5}$$

Table 5 presents the performance of each model on the training and validation sets after 100 epochs under both evaluation metrics. All models performed well under both evaluation metrics, and their performances on the training and validation sets were similar, indicating strong generalization ability to new data not seen during training. Overall, the models trained with the SDF method outperformed those trained with the BRN method, as shown in Figure 12, with the RMSE of the BRN models ranging from 1.1765 m/s to 2.0138 m/s and that of the SDF models ranging from 0.7966 m/s to 0.9037 m/s. Correspondingly, the R^2 scores of the BRN models on the training and validation sets were distributed between 0.8636 and 0.9508, while those of the SDF models were distributed between 0.9716 and 0.9776. Moreover, the model performance improved gradually with the increase in convolutional layers and Vgg-CFD-19 achieved the best performance under both methods.

Table 5. Training and validation results for each model.

			Vgg-CFD-11	Vgg-CFD-13	Vgg-CFD-16	Vgg-CFD-19
BRN	Train	RMSE	2.0138	1.2269	1.2066	1.2472
		R^2	0.8591	0.9473	0.9489	0.9460
	Validation	RMSE	1.9765	1.2349	1.2543	1.1765
		R^2	0.8636	0.9469	0.9456	0.9508
SDF	Train	RMSE	0.9037	0.8517	0.8300	0.8391
		R^2	0.9716	0.9747	0.9761	0.9755
	Validation	RMSE	0.8708	0.8579	0.8374	0.7966
		R^2	0.9731	0.9742	0.9752	0.9776

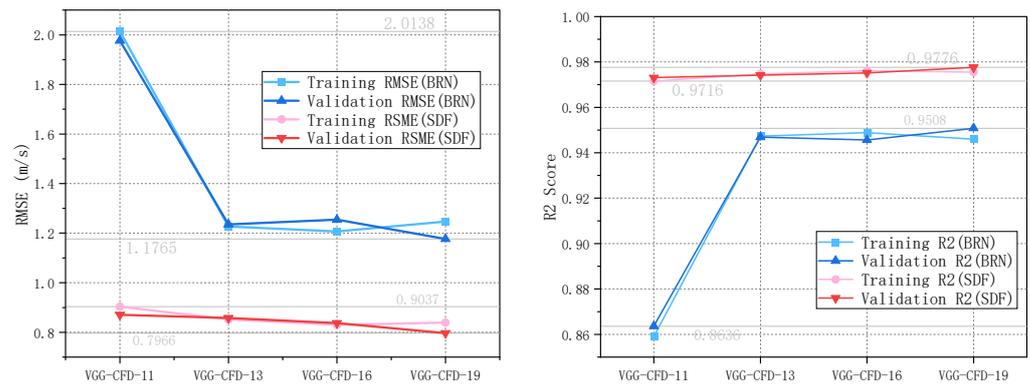


Figure 12. Training and validation results for each model.

3.2. Performance on the Testing Set

In the previous model training, although the validation set data did not appear in the model training process, both the validation set and the training set were derived from the selected sample cases. To better demonstrate the generalization ability of the model, we introduced a randomly selected case that did not participate in the training for testing. The shape parameters of the test set are shown in Table 6, and the wind field was calculated using OpenFOAM with the wind field calculation method mentioned before, as shown in Figure 13.

Table 6. Configuration of the test case.

Parameters (m)	Building A		Building B		Street Valley
	Height	Width	Height	Width	Width
	20	20	9	12	15

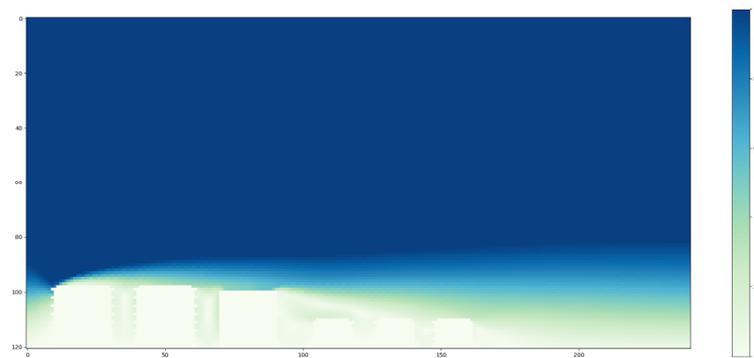


Figure 13. OpenFOAM calculation results of the test case.

Figure 14 shows the wind environment prediction results of four models with two methods. The results and errors from OpenFOAM calculation are expressed in cloud maps. Both methods reflected the basic characteristics of the flow field. However, the BRN-based models trained with the SDF-based method made relatively poorer predictions compared to the models trained with the BRN-based method.

Figure 15 and Table 7 display the overall prediction performance of each model, from which we can observe that the best-performing model was Vgg-CFD-16 based on the SDF method, with an R^2 value of 0.8455 and an RMSE of 2.2345 m/s.

To further understand the prediction performance of the models at different locations, a vertical-section analysis was conducted 5 m in front of the building ($x = 5$), at the midpoint of the street canyon ($x = 98$), and 5 m behind the building ($x = 167$). As shown in Figure 16, line graphs of the vertical wind field at these positions were created, where the result

calculated with OpenFOAM was represented using the red solid line and the four models trained with the BRN methods were represented using blue dotted lines of different depths, while the four models trained with the SDF method were represented using green dotted lines. In general, the prediction results of all models were found to follow the trend of the OpenFOAM calculation results, with good fitting performance observed in the upwind direction, slightly poorer performance in the middle of the street canyon, and the poorest performance in the downwind direction. It is difficult to predict these areas probably due to the influence of turbulence, as these areas experience more dramatic fluctuations in the wind field, and there is some degree of uncertainty in the distribution of the wind field, making it difficult for the models to identify regular patterns. Additionally, there was a certain trend toward a right offset of the blue and green dotted lines compared to the red solid line in the chart—especially evident in the height range of over 40 m. In other words, all models tended to overestimate wind speed values in the higher parts, probably due to the deviation of the distribution of the training set. In the calculation of the previous training case, to obtain the wind environment characteristics of all buildings, the output height of the case result was set at 1.5 times the maximum height of the potential buildings in the case (120 m). This causes the high-wind-speed areas above the buildings to occupy a larger proportion when the buildings are low in height, resulting in an overall bias toward higher wind speed distribution in the training set, ultimately affecting the training direction of the models.

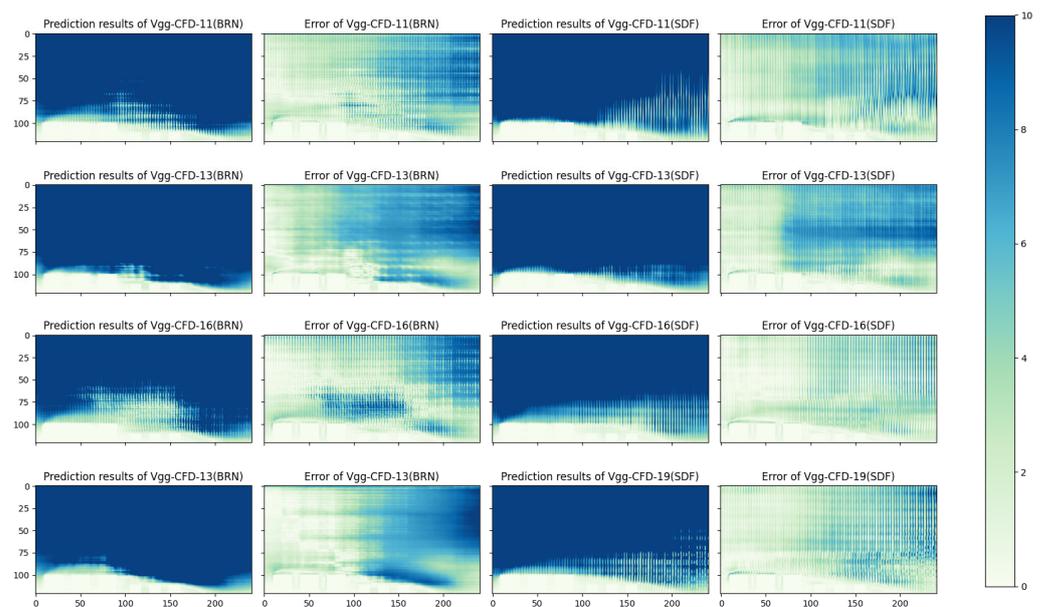


Figure 14. Prediction and error analysis of each model group.

Table 7. Testing results for each model.

		Vgg-CFD-11	Vgg-CFD-13	Vgg-CFD-16	Vgg-CFD-19
BRN	RMSE	3.1242	3.6620	3.3057	3.5397
	R2 score	0.6980	0.5851	0.6619	0.6123
SDF	RMSE	3.2054	3.4129	2.2345	2.7167
	R2 score	0.6821	0.6396	0.8455	0.7717

3.3. Program Packaging

To encourage architects to quantify the environmental performance of buildings during the design process, our team developed an urban comprehensive performance analysis platform called URBAN NEURAL, based on Grasshopper. The platform uses neural network models to quickly calculate the environmental performance in various urban

settings, as shown in Figure 17. URBAN NEURAL is a scalable, performance-oriented architecture, and modules can calculate environmental performance by changing the model files. In previous tests, we found that the Vgg-CFD-19 model performed better in all aspects in the models represented using the SDF method. Therefore, it was embedded in the URBAN NEURAL-CFD module. After testing, we found that the tools take less than 5 min, which is about 50 times faster than traditional CFD calculations.

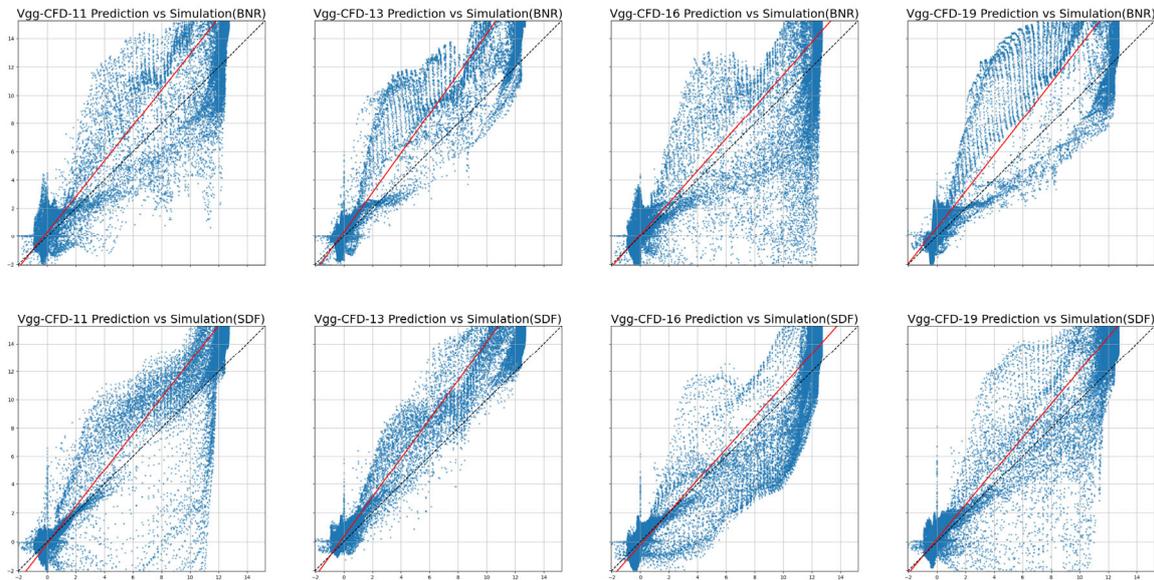


Figure 15. Scatter plots of the prediction errors of each model group on the test case.

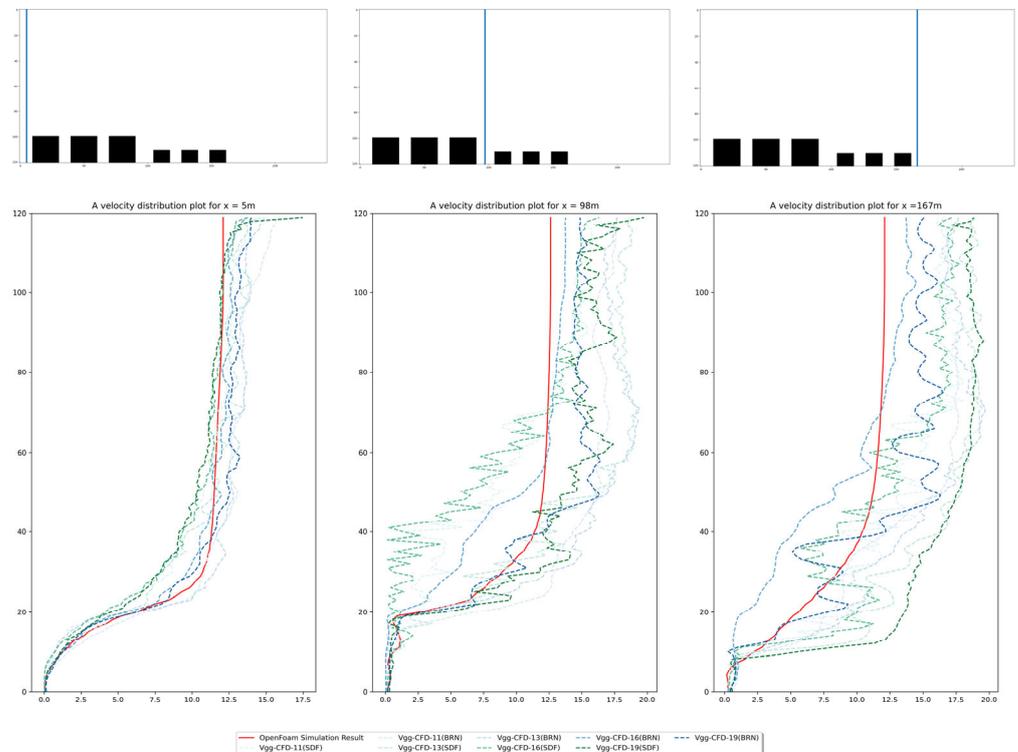


Figure 16. Prediction of the vertical wind speed and comparison of the calculated values at different locations.

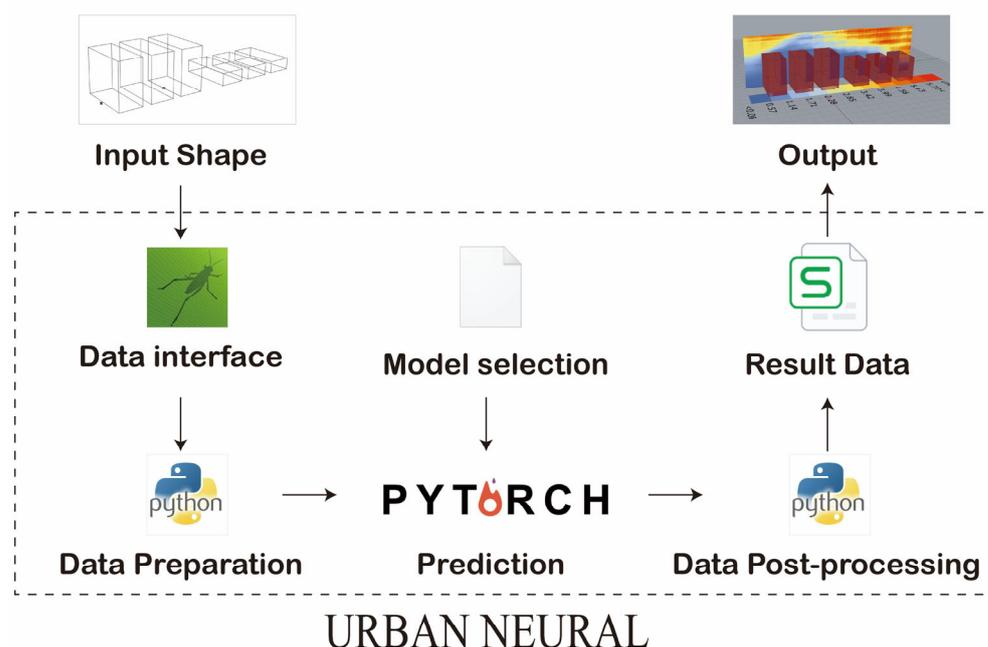


Figure 17. URBAN NEURAL workflow.

4. Discussion and Limitation

We conducted experiments on four models with two representation methods using validation and testing sets. Overall, all models learned spatial characteristics of the flow field to some extent and can predict the steady-state flow field in various neighborhoods.

4.1. Diversity of Scenarios, Accuracy, and Sample Size

Neural network models generally require a larger sample size to achieve the same accuracy when applied to more diverse scenarios. In addition, there is a positive correlation between a model's accuracy and sample size, with diminishing marginal returns [36]. Therefore, when the computing power is limited, these three factors constrain each other and are contradictory.

Although the requirements for urban designers and architects, especially in the early stages of design, are relatively simple, the changes in urban space are complex. Buildings unfold freely in a homogeneous space, and it is difficult to describe a complex urban space solely through parameters such as length, width, and height. The BNR and SDF methods described in this study both use a minimum unit to partition urban space, and the numerical value of the unit represents the spatial characteristics of the unit's location. Similarly, the finite element method commonly used in solving the physical environment discretizes the continuous space, divides it into finite elements, and then approximates the solution through numerical simulation. This method can be extended to a three-dimensional space [2] and is suitable for most scenarios in the early stage of design.

Describable scenes require a sufficient amount of samples to ensure the accuracy of the model. In related literature research, Guo et al. [2] used a huge training set of 100,000 cases. Mokhtar et al. [9] used 2800 cases, with 19,200 pairs of image data sampled at different heights, and Duering et al. [15] took 6 months for data preparation and training. The common feature of these studies is the establishment of the relationship between the spatial field and flow field. The model's input and output ports typically have matching dimensions. A single datum usually has a large number of dimensions, typically exceeding 512×512 , for both input and output parameters. This significantly increases the complexity of the model, the amount of data, and the time required for training.

In this article, to investigate the relationship between the field containing a specific point and the corresponding wind speed at that point, we shifted our focus from the

relationship between different fields to the perspective of individual points, which brings two advantages.

- (1) Neural network models can be effectively trained with small-scale data. Each wind farm corresponds to tens of thousands of data points, resulting in a significant increase in the amount of data. Furthermore, non-uniform sampling was carried out according to the research area of interest, which skewed resources toward areas of high importance, allowing more complete use of data and computational resources. However, these practices also present risks as the data may be homogeneous, causing overfitting of the neural network models. In this study, the model exhibited some overfitting, but it was not severe overall. Taking the best-performing model on the testing set, which is the Vgg-CFD-19 model under the scalar distance function (SDF) representation, as an example, it had an R2 score of 0.9776 and an RMSE of 0.7966 m/s on the training dataset, and a slight decrease was observed on the validation set with an R2 score of 0.7717 and an RMSE of 2.7167 m/s. To mitigate these effects in future work, a possible solution is increasing the training set size.
- (2) The calculation of the proxy model can be unconstrained with the intrinsic size of the input and output of the neural network model. In previous studies, due to the relationship between fields established with the model, the neural network model determined the size and dimension of the fields through a predefined definition and the scene setting in the model calculation needed to follow this size, which could not be changed. For example, Mokhtar et al. [9] used the pix2pix model with an input size of 1024×1024 ; Guo et al. [2] used input and output sizes of 256×128 . In this study, the neural network models relied only on the environments surrounding the measurement points, enabling the models to accept an arbitrary number from areas of any size or shape, providing more versatility in the applicability of the model.

4.2. Network Architecture and Geometric Representation

This article compared the prediction performance of two representation methods and four deep convolutional neural network models, and the results presented two features.

- (1) The SDF representation models were generally superior to the BNR representation models. For both representation methods, the performance of each training model on the corresponding training, testing, and validation datasets was clearly stratified, with SDF representation consistently outperforming BNR representation. The reason may be that compared with the BNR representation method, the SDF representation method has global information, which can reflect the distance and shape of the nearest obstacles in the entire space to a certain extent in any spatial slice. This method of increasing the density of model input information may become a direction for further research in the future.
- (2) The depth of convolutional neural networks is positively correlated with their prediction accuracy. In terms of the performance of the four models on the dataset, as the depth of the model increased, the predictive performance of the model gradually improved, consistent with the research on deep learning literature [37]. However, this trend was not stable. For example, in the BNR representation method, the performance of Vgg-CFD-16 was inferior to that of Vgg-CFD-13. This may be due to the randomness of the model training process, which may result in decreased accuracy of the deeper network. However, negative effects such as gradient explosion caused by the increase in the depth of the neural network cannot offset the positive effects it brings. Therefore, in future research, ResNet or hyperparameter optimization methods can be used for model optimization.

4.3. URBAN NEURAL Platform

Traditionally, to quantitatively judge building environmental performance, designers use software for modeling and calculation. However, using different software for modeling and calculating can be challenging due to varying data formats and the knowledge of

physical laws required. The fast fluid dynamics method (FFD), another popular fast computational approach, similar to machine learning approaches, also entails a trade-off between sacrificing a certain level of accuracy and achieving speed enhancements. However, these two methods diverge in their technical trajectories. However, they differ in their technical paths. FFD model simplifies CFD algorithms based on physics principles, while machine learning methods primarily rely on data-driven approaches to uncover underlying patterns. This results in an advantage for machine learning methods, specifically in dealing with complex flow fields or multi-field coupling problems, as they demonstrate greater feasibility. Despite being in its early stages, the emerging research demonstrates its tremendous potential. Therefore, we propose the URBAN NEURAL platform based on Grasshopper to achieve fast calculation of various urban environmental performances using neural network models. The platform simplifies the calculation process by using a three-step process comprising data pre-processing, neural network model calculation, and data post-processing to obtain environmental performance parameters. These steps are automatically completed within the platform to reduce the learning cost and use threshold for designers. The Vgg-CFD-19 model, which showed the best performance, is integrated into the platform as the URBAN NEURAL-CFD module, and has greater calculation efficiency compared to traditional CFD calculations. In the future, we plan to further develop this framework to reduce the cost of designing low-carbon passive structures and promote these practices among designers.

4.4. Limitation and Future Work

This paper explores using a small amount of sample data to train an approximate model for predicting non-uniform steady-state flow fields based on convolutional neural networks. The model achieves a certain level of predictive performance, an R^2 of 0.8455, and an RMSE of 2.2345 m/s on the testing dataset. Compared to previous research [2,10,22,23], we significantly reduced the number of cases required for training at the cost of slightly lower accuracy. However, there are two aspects worth noting regarding these results.

- (1) The relative error in predicting weak-wind-speed regions is large. In urban wind environment issues, we are more concerned about the wind environment changes in the pedestrian layer near the ground, where wind speeds are usually low. In Figure 15, we can see that although the absolute error generated in this area is small, the error percentage is relatively large. This will make it difficult to guide the optimization of the design in the near-surface region with the help of the predicted data. This situation also appeared in other previous studies. For example, in the study by Tanaka, the area with relative errors greater than 50% was also concentrated in the building wake area [22]. This may be related to the loss function defined in the model. In this paper, the mean squared error is used as the loss function, because of which, the model tends to reduce the absolute value of the error. In future work, we can try to reduce the impact of this factor by increasing the weight of wind speed errors near the ground.
- (2) There is a lack of quantitative measurement relationships between scenario diversity, accuracy, and the required size of the training data. According to the no-free-lunch theorem, it is difficult to obtain a model that is optimal in all three aspects. As the amount of data increases, the accuracy and applicability of the obtained model will further increase. To achieve a balance between applicability, economy, and accuracy, we need to understand the quantitative relationship between these three factors to guide the construction of future neural-network-based models for assessing building environmental performance as a substitute.

5. Conclusions

In this paper, we presented a deep convolutional neural network method trained on small-scale data for predicting the steady-state flow field in urban blocks. We constructed a neural network model that maps the surrounding environment of a point to its wind velocity performance, using a dataset generated through Latin hypercube sampling. We

also discussed the performance differences of two geometric representations and different network structures, ultimately embedding the best-performing model into the URBAN NEURAL platform for a comprehensive city performance analysis.

The contributions and conclusions of this paper are as follows:

- (1) We developed a deep convolutional neural network alternative model that achieves high prediction accuracy for steady-state flow fields using small-scale data. Compared with previous studies, this model achieved a better balance of applicability, affordability, and precision.
- (2) The signed distance function data representation outperformed the Boolean network representation.
- (3) Vgg-CFD-19 showed better performance, and the accuracy of the network was positively correlated with the number of convolutional neural network layers.
- (4) In the future, with further research, URBAN NEURAL will gradually become a more versatile urban performance analysis platform.

Author Contributions: Conceptualization, G.Z.; methodology, G.Z.; writing—original draft, G.Z.; writing—review and editing, G.Z., X.X. and L.Y.; software, J.F.; supervision, L.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Department of Science and Technology of Guangdong Province grant number [2023A1515011647], and the Science, Technology and Innovation Commission of Shenzhen Municipality grant number [JCYJ20180305125219726].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data will be made available on request.

Acknowledgments: The authors would like to thank the anonymous referees and the editors for their helpful suggestions on the earlier draft of the paper.

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

References

1. Shao, X.; Liu, Z.; Zhang, S.; Zhao, Z.; Hu, C. PIGNN-CFD: A physics-informed graph neural network for rapid predicting urban wind field defined on unstructured mesh. *Build. Environ.* **2023**, *232*, 110056. [\[CrossRef\]](#)
2. Guo, X.; Li, W.; Iorio, F. Convolutional Neural Networks for Steady Flow Approximation. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 481–490. [\[CrossRef\]](#)
3. Zaghoul, M. Machine-Learning Aided Architectural Design-Synthesize Fast CFD by Machine-Learning. Ph.D. Thesis, ETH Zurich, Zurich, Switzerland, 2017. [\[CrossRef\]](#)
4. Xu, X.; Gao, Z.; Zhang, M.; Hammond, D.S.; Chapman, L.; Thornes, J.E. A review of simplified numerical approaches for fast urban airflow simulation. *Build. Environ.* **2023**, *234*, 110200. [\[CrossRef\]](#)
5. Stam, J. Real-Time Fluid Dynamics for Games. In Proceedings of the Game Developer Conference, San Jose, CA, USA, 4–8 March 2003.
6. Zuo, W.; Chen, Q. Fast and informative flow simulations in a building by using fast fluid dynamics model on graphics processing unit. *Build. Environ.* **2010**, *45*, 747–757. [\[CrossRef\]](#)
7. Hang, J.; Li, Y. Wind Conditions in Idealized Building Clusters: Macroscopic Simulations Using a Porous Turbulence Model. *Bound. Layer Meteorol.* **2010**, *136*, 129–159. [\[CrossRef\]](#)
8. Mao, Y.-F.; Li, Z.; He, Y.-L.; Tao, W.-Q. Numerical investigation of dust sedimentation effects on wall adsorption of indoor SVOC by the immersed boundary-lattice Boltzmann method. *Build. Environ.* **2020**, *180*, 106974. [\[CrossRef\]](#)
9. Mokhtar, S.; Sojka, A.; Davila, C.C. Conditional generative adversarial networks for pedestrian wind flow approximation. In Proceedings of the 11th Annual Symposium on Simulation for Architecture and Urban Design, Online, 25–27 May 2020; pp. 1–8.
10. Calzolari, G.; Liu, W. Deep learning to replace, improve, or aid CFD analysis in built environment applications: A review. *Build. Environ.* **2021**, *206*, 108315. [\[CrossRef\]](#)
11. Brunton, S.L.; Noack, B.R.; Koumoutsakos, P. Machine Learning for Fluid Mechanics. *Annu. Rev. Fluid Mech.* **2020**, *52*, 477–508. [\[CrossRef\]](#)
12. Gan, V.J.L.; Wang, B.; Chan, C.M.; Weerasuriya, A.U.; Cheng, J.C.P. Physics-based, data-driven approach for predicting natural ventilation of residential high-rise buildings. *Build. Simul.* **2022**, *15*, 129–148. [\[CrossRef\]](#)

13. Zhou, Q.; Ooka, R. Influence of data preprocessing on neural network performance for reproducing CFD simulations of non-isothermal indoor airflow distribution. *Energy Build.* **2021**, *230*, 110525. [[CrossRef](#)]
14. Musil, J.; Knir, J.; Vitsas, A.; Gallou, I. Towards Sustainable Architecture: 3D Convolutional Neural Networks for Computational Fluid Dynamics Simulation and Reverse Designworkflow. *arXiv* **2019**, arXiv:1912.02125. [[CrossRef](#)]
15. Duering, S.; Chronis, A.; Koenig, R. Optimizing Urban Systems: Integrated Optimization of Spatial Configurations. In Proceedings of the 11th Annual Symposium on Simulation for Architecture and Urban Design, Vienna, Austria, 25–27 May 2020; p. 7.
16. Raissi, M.; Yazdani, A.; Karniadakis, G.E. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* **2020**, *367*, 1026–1030. [[CrossRef](#)]
17. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics Informed Deep Learning (Part I): Data-driven Solutions of Nonlinear Partial Differential Equations. *arXiv* **2017**, arXiv:1711.10561.
18. Cai, S.; Mao, Z.; Wang, Z.; Yin, M.; Karniadakis, G.E. Physics-informed neural networks (PINNs) for fluid mechanics: A review. *Acta Mech. Sin.* **2021**, *37*, 1727–1738. [[CrossRef](#)]
19. Dai, X.; Liu, J.; Zhang, X.; Chen, W. An artificial neural network model using outdoor environmental parameters and residential building characteristics for predicting the nighttime natural ventilation effect. *Build. Environ.* **2019**, *159*, 106139. [[CrossRef](#)]
20. Zhou, Q.; Ooka, R. Comparison of different deep neural network architectures for isothermal indoor airflow prediction. *Build. Simul.* **2020**, *13*, 1409–1423. [[CrossRef](#)]
21. Tanaka, H.; Matsuoka, Y.; Kawakami, T.; Azegami, Y.; Yamamoto, M.; Ohtake, K.; Sone, T. Optimization calculations and machine learning aimed at reduction of wind forces acting on tall buildings and mitigation of wind environment. *Int. J. High Rise Build.* **2019**, *8*, 291–302. [[CrossRef](#)]
22. Huang, C.; Zhang, G.; Yao, J.; Wang, X.; Calautit, J.K.; Zhao, C.; An, N.; Peng, X. Accelerated environmental performance-driven urban design with generative adversarial network. *Build. Environ.* **2022**, *224*, 109575. [[CrossRef](#)]
23. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015), San Diego, CA, USA, 7–9 May 2015; pp. 1–14.
24. Wei, X. Visual Analysis of Fine-Grained Images Based on Deep Learning. Ph.D. Thesis, Nanjing University, Nanjing, China, 2018.
25. Bhatnagar, S.; Afshar, Y.; Pan, S.; Duraisamy, K.; Kaushik, S. Prediction of aerodynamic flow fields using convolutional neural networks. *Comput. Mech.* **2019**, *64*, 525–545. [[CrossRef](#)]
26. Zhong, G. Convolutional Neural Network Model to Predict Outdoor Comfort UTCI Microclimate Map. *Atmosphere* **2022**, *13*, 1860. [[CrossRef](#)]
27. Singh, M.M.; Singaravel, S.; Geyer, P. Machine learning for early stage building energy prediction: Increment and enrichment. *Appl. Energy* **2021**, *304*, 117787. [[CrossRef](#)]
28. Deutsch, J.L.; Deutsch, C.V. Latin hypercube sampling with multidimensional uniformity. *J. Stat. Plan. Inference* **2012**, *142*, 763–772. [[CrossRef](#)]
29. Shen, L.; Han, Y. Comparative Analysis of Machine Learning Models Optimized by Bayesian Algorithm for Indoor Daylight Distribution Prediction. In Proceedings of the Planning Post Carbon Cities: 35th PLEA Conference on Passive and Low Energy Architecture, A Coruña, Spain, 1–3 September 2020; Universidade da Coruña: Coruña, Spain, 2020.
30. Meroney, R.; Ohba, R.; Leitl, B.; Kondo, H.; Grawe, D.; Tominaga, Y. Review of CFD Guidelines for Dispersion Modeling. *Fluids* **2016**, *1*, 14. [[CrossRef](#)]
31. Yoshie, R.; Mochida, A.; Tominaga, Y.; Kataoka, H.; Harimoto, K.; Nozu, T.; Shirasawa, T. Cooperative project for CFD prediction of pedestrian wind environment in the Architectural Institute of Japan. *J. Wind Eng. Ind. Aerodyn.* **2007**, *95*, 1551–1578. [[CrossRef](#)]
32. Hammond, D.S.; Chapman, L.; Thornes, J.E. Roughness length estimation along road transects using airborne LIDAR data. *Meteorol. Appl.* **2012**, *19*, 420–426. [[CrossRef](#)]
33. Wieringa, J. New revision of Davenport roughness classification. In Proceedings of the 3EACWE, Eindhoven, The Netherlands, 2–6 July 2001; pp. 285–292.
34. Umetani, N.; Bickel, B. Learning three-dimensional flow for interactive aerodynamic design. *ACM Trans. Graph.* **2018**, *37*, 1–10. [[CrossRef](#)]
35. Kingma, D.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980. [[CrossRef](#)]
36. Ng, W.; Minasny, B.; Mendes, W.d.S.; Dematté, J.A.M. The influence of training sample size on the accuracy of deep learning models for the prediction of soil properties with near-infrared spectroscopy data. *Soil* **2020**, *6*, 565–578. [[CrossRef](#)]
37. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.