



A Rapid Wind Velocity Prediction Method in Built Environment Based on CycleGAN Model

Chuheng Tan¹(✉) and Ximing Zhong²(✉)

¹ The Bartlett School of Architecture, University College London, 22 Gordon Street, London, UK

chuheng.tan.21@ucl.ac.uk

² Aalto University Finland Espoo, 02150 Espoo, Finland

ximing.zhong@aalto.fi

Abstract. Although the wind microclimate and wind environment play important roles in urban prediction, the time-consuming and complicated setup and process of wind simulation are widely regarded as challenges. There are several methods to use deep learning (DL) models for wind speed prediction by labeling pairs of wind simulation dataset samples. However, many wind simulation experiments are needed to obtain paired datasets, which is still time-consuming and cumbersome. Compared with previous studies, we propose a method to train a DL model without labelling paired data, which is based on Cycle Generative Adversarial Network (cycleGAN). To verify our hypothesis, we evaluate the results and process of the pix2pix model (requires paired datasets) and cycleGAN (does not requires paired datasets), and explore the difference of results between these two DL models and professional CFD software. The result shows that cycleGAN can perform as well as pix2pix in accuracy, indicating that some random city plans image samples and random wind simulation samples can train surrogate models as accurate as labelled DL methods. Although the DL method has similar results to the professional CFD method, the details of the wind flow results still need improvement. This study can help designers and policymakers to make informed decisions to choose DL methods for real-time wind speed prediction for early-stage design exploration.

Keywords: Deep learning · Wind velocity prediction · Pix2pix · CycleGAN

1 Introduction

The Wind environment is an important factor in urban microclimate studies [1]. Since 1930s, the Navier–Stokes equations have been used for calculating fluid dynamics problems, which enable people to use mathematical equations to simulate the principles of fluid mechanics [2]. The development of computer technology increased the development of three-dimensional methods. Until now, Computational Fluid Dynamics (CFD) simulation is regarded as one of the most popular and useful methods for urban wind

Zhong: Contributed equally to this work and should be considered the co-first author.

© The Author(s) 2023

P. F. Yuan et al. (eds.), *Hybrid Intelligence*, Computational Design and Robotic Fabrication, https://doi.org/10.1007/978-981-19-8637-6_22

environment studies but is usually complicated and time-consuming, which is widely regarded as a challenge [3]. In 1999, the advent of Fast Fluid Dynamics (FFD) proposed by Jos Stam [4] addresses this challenge to some extent [5]. FFD is a technique for solving the incompressible Navier–Stokes equations and it is developed for real-time fluid visualization in the video and gaming industry, consequently, which has the advantage of being very fast to run and simple to code [6]. Since the 2000s, Deep Learning (DL) methods have been implemented in predicting the wind environment, providing the possibility to analyze or predict the urban wind environment much faster [7]. Although some research on simulating and predicting the urban wind environment based on GAN and FFD has been proposed [5], a large number of wind simulation experiments are needed to obtain paired data sets, which is a complex and time-consuming process for data preparation and labeling [8]. CycleGAN is a new technique that involves the automatic training of image-to-image translation models, without paired examples. It has the advantage of training models in an unsupervised manner using a collection of images from both source and target domains that do not need to be related in any way [9]. How to achieve a workflow that can produce wind speed predictions without preparing paired datasets remains a challenge.

The main purpose of this article is to find a more convenient DL workflow, rather than improve the accuracy of existing simulation software. Compared with the previous study, we choose the cycleGAN model, which doesn't require paired and labelled datasets. Designers input only random city samples, and random wind simulation samples, which are not matched. To verify our hypothesis, we evaluate the results accuracy of two different GAN models, pix2pix and cycleGAN, aiming to find a fast and easy workflow to generate high-quality results for city block wind prediction. Besides, we compare the results with CFD simulation results concurrently to discover the advantages and limitations of this method.

2 Related Work

Many studies successfully used machine learning and the DL method to investigate wind characteristics and wind effects on buildings and the urban environment (e.g., wind flow, wind pressure, and wind-induced responses) [8]. The large full flow fields data of CFD simulations has sparked interest in applying deep neural networks (DNNs) for fast approximations to computational fluid dynamics. DNNs were firstly applied to surrogate modelling and realize design exploration [10]. Convolutional neural networks (CNNs) can perform better than DNNs in the design exploration domain because they represent non-linear input and output functions while extracting spatial relationship [5], such as Jin's prediction of the velocities filled around a circuit cylinder [11].

The advent of Generative Adversarial Networks (GANs) greatly reduces the calculation cost and time and produces minor error results by learning the characteristics of paired samples [12, 13]. Kim develops a kind of GAN model, called Generative Adversarial Imputation Network (GAIN) to impute the unmeasured velocities around buildings [8]. He Yi develops a hybrid framework for rapid evaluation of wind velocity around buildings through parametric design, CFD simulation and machine learning on Pix2pix [14]. Mokhtar used cGAN and Pix2pix for pedestrian wind comfort estimation [15].

The studies above highlight the potential of deep learning networks in learning and predicting the wind environment. As discussed in the introduction section, labeling datasets is a cumbersome process, a large number of paired maps of urban block prototype plans and corresponding wind simulation results sometimes are hard to prepare. CycleGAN model is a kind of unsupervised model for image-to-image translation that doesn't require paired datasets, which makes it possible for faster dataset collection [9].

In contrast with previous work, our focus is to verify whether unsupervised methods that don't need labeled datasets can also realize wind prediction. In this study, we compare the performance of two different GAN models, a supervised model (Pix2pix) and an unsupervised one (cycleGAN) in the DL method for wind velocity simulation. The aim is to find a faster workflow to generate datasets without labelling and obtain better training results in a shorter time. Compared with the supervised model, this workflow enables designers to train the unsupervised model for wind environment prediction faster and easier.

3 Method

The research workflow of this paper is shown in Fig. 1. The process can be divided into 3 steps: (1) dataset preparation, (2) pix2pix model and cycleGAN model training, (3) comparison of the results from these two models. We use Houdini to generate city block plans and perform wind simulation in step2, and train the current DL model in Google Colab in step2. Finally, we observe the training loss and FID value of two models in step 3, the focus of our study, aims to evaluate the performance of these two models and to find a faster workflow for city wind velocity prediction.

3.1 Data Preparation for the Pix2pix Model and CycleGAN Model

The main difference between the pix2pix model and the cycleGAN model is the use of labelled datasets. The former uses labelled input and output data, while the latter does not [16]. Pix2pix is a kind of supervised model that's defined by its use of labelled input and output datasets, it is composed of a generator and a discriminator. These datasets are designed to train or "supervise" algorithms into classifying data or predicting outcomes accurately [17]. A cycleGAN model works on its own to discover the hidden pattern in the dataset without the need for human intervention. It is trained in an unsupervised manner using a collection of images from the source and target domain without a one-to-one mapping between the source and target domain that does not need to be related or paired in any way [9]. Compared to supervised models such as pix2pix, it is more convenient and faster to create a large dataset, and we don't need to make paired datasets samples and label them.

There are two sets of datasets for these two models, datasets 1 (D1) is for pix2pix model training, which contains input (train A) and output (train B) datasets. Train A are city block plans and train B are their corresponding wind simulation results, the A and B images are paired and matched. Datasets 2 (D2) is for cycleGAN model training, which also contains train A and train B, but they are random and not matched.

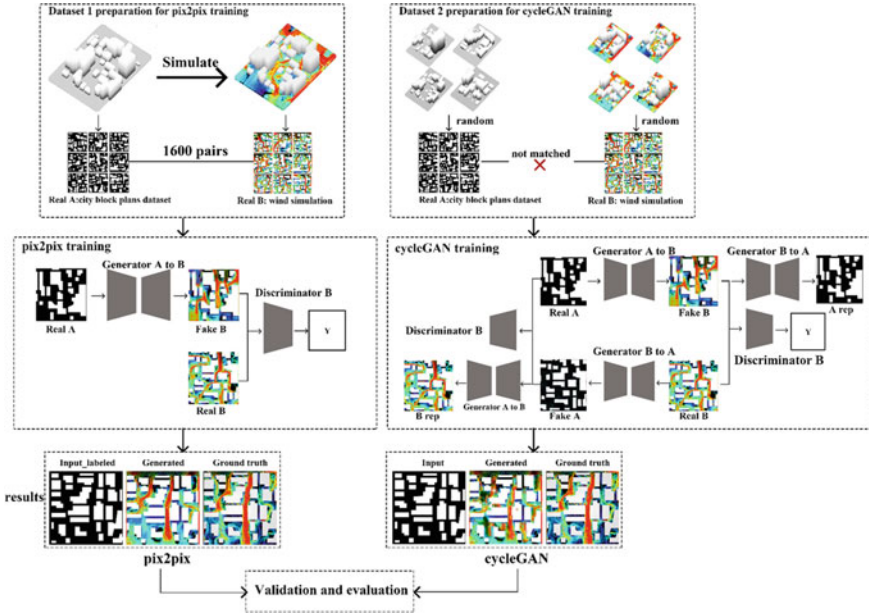


Fig. 1. The whole workflow of comparing two DL models for wind prediction in this study

In this study, we choose Houdini parametric modelling to prepare city blocks and generate wind velocity simulation, as Houdini has a good synergy in modelling and wind simulation for our comparative experiment. Firstly, we use Houdini parametric modelling method attached to the PDG system to generate city blocks prototype plans. The PDG system provides a rich set of stock nodes to enhance productivity, which can generate a great number of datasets automatically in a short time [18]. Every generated block plan was limited to a $200\text{ m} \times 200\text{ m}$ square, which was based on typical dimensions and densities of urban contexts [15]. And in each block, the buildings were set up with heights ranging from 9 to 100 m, and lengths and widths ranging from 12 to 60 m.

Secondly, each city block is entered into the Houdini wind simulation system automatically, which has a good synergy. This simulation method is a kind of FFD method, which has been verified as a relatively fast and high-accuracy method in 2015 [19] and it is successfully applied in much parametric design research, such as Nodado et al.'s wind-induced architectural systematization [20]. It can build different models and generate corresponding wind environment analysis charts in real-time without manually exporting the model for wind simulation in other software. To set up the wind test environment, the minimum meshing cell size is defined as 3 m for the models, which can take a better balance between computational feasibility and a low average error. The wind velocity was set as 5 m/s at a 10 m reference height for all plans. This wind velocity was commonly observed in most urban area [15]. As for the datasets for the pix2pix model, the input images (city blocks plans) and output images (wind speed simulation) need to be sorted consistently and set in the same size for matching into pairs in the machine learning process. For architects, this process may be tedious and error-prone.

Finally, for training the Pix2pix model, we generated 1600 paired city block plans and corresponding wind speed simulation images, the colormap of wind analysis can be converted to label information. And for training the cycleGAN model, we generate 1600 city block plans and 1600 wind simulation images that aren't matched. Figure 2 datasets samples for pix2pix model and cycleGAN model separately. The dataset samples are shown. For curating the datasets, we divided the dataset into 90% training and 10% testing sets, thus we used 1440 in training and 160 in testing, according to the sample density and proportion of the dataset.

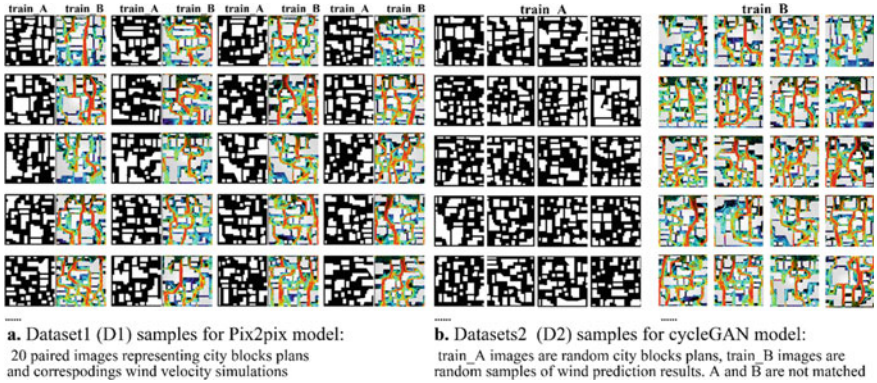


Fig. 2. Datasets samples for the pix2pix model and cycleGAN model separately

3.2 DL Model Training

In this section, we train the pix2pix model and cycleGAN to find an accurate and faster wind simulation and prediction workflow. The two processes are set up and implemented in Google Colab, which provides a runtime fully configured for deep learning and free-of-charge access to a robust GPU, and it is convenient to write and execute code [21].

Pix2pix model is developed based on a conditional generative adversarial network (cGAN) to learn a function to map from an input image to an output image. The network is made up of two main pieces, the Generator, and the Discriminator. The Generator transforms the input image to get the output image [22]. In this study, we use the pix2pix proposed by Phillip Isola in 2017 [22], the input and output resolution are set to 256×256 pixels, a 50 m receptive field for the discriminator, with a learning rate of 0.0002 and generator adversarial to L1 loss ratio of 1 to 100, and all experiments were trained for a total of 200 epochs. After every epoch, the trained weights were saved to monitor the progress of training. Figure 3a shows the process of pix2pix model training.

In this study, we use the cycleGAN model proposed by Jun-Yan Zhu and Taesung Park in 2017 [17]. A cycleGAN model is composed of 2 GANs, making it a total of 2 generators and 2 discriminators. One generator transforms city block plans into city blocks with wind speed simulation result, and the other transform wind results into city block plans. In the case of cycleGAN, a generator gets additional feedback from the

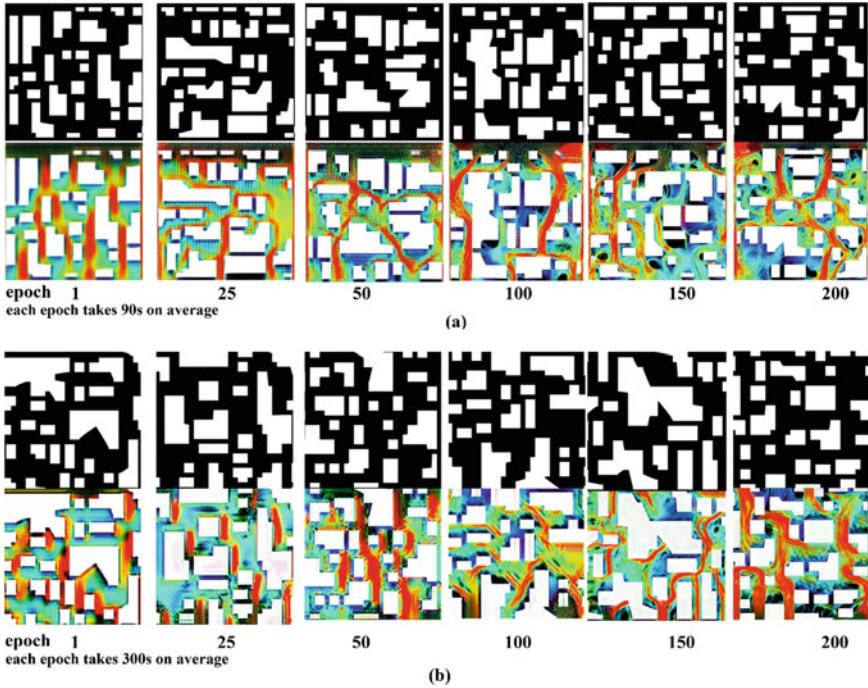


Fig. 3. a pix2pix training process b cycleGAN training process during epoch 200

other generator. This feedback ensures that an image generated by a generator is cycle consistent, meaning that applying consecutively both generators on an image can yield a similar image.

To ensure the same experimental conditions, in this study, we set the same environment and parameters for the cycleGAN model as for the pix2pix model. The image of training dataset A and B are converted to 256×256 pixels, with the learning rate of 0.0002 for a total of 200 epochs. When training the discriminator, the loss is divided by 2. The weights are initialized with a Gaussian distribution with a mean of 0 and a standard deviation of 0.02. Every epoch the training set is shuffled and partitioned into subsets the size of the minibatch. After every epoch, the trained weights were saved to monitor the progress of training. Figure 3b shows the process of cycleGAN training.

3.3 Performance Evaluation Method and Criteria

Firstly, we evaluate the stability and accuracy of these two models by mainly observing the Training loss and the Fréchet inception distance (FID). We compare the training loss graphs to observe the loss decreasing and convergence of the two models as shown in Fig. 5a. For clearer image contrast, we reparametrize the value of the diagram. The FID is a metric used to assess the quality of images created by a generative model, especially in GAN [23]. It is the current standard metric for assessing the quality of GANs as of 2020, which has been used to measure the quality of many recent GANs. The FID is

improved on the IS by actually comparing the statistics of generated samples to real samples. It compares the distribution of generated images with the distribution of real images that were used to train the generator [24]. The FID score is then calculated using the following equation in this study:

$$FID = ||\mu - \mu_w||_2^2 + \text{Tr}\left(\Sigma + \Sigma_w - 2\left(\Sigma^{\frac{1}{2}}\Sigma_w\Sigma^{\frac{1}{2}}\right)^{\frac{1}{2}}\right) \quad (1)$$

The FID metric is the squared Wasserstein metric between two multidimensional Gaussian distribution: $\mathcal{N}(\mu_w, \Sigma_w)$, the distribution of the neural network features of the images generated by the GAN and $\mathcal{N}(\mu, \Sigma)$ the distribution of the same neural network features from the real images used to train the GAN [24].

A lower FID indicates better-quality images, conversely, a higher score indicates a lower-quality image and the relationship may be linear.

Secondly, although the FFD simulation method in Houdini is a practical and rapid way to generate large datasets, the accuracy may not be as well as the professional CFD simulation software. Considering more convincing results, we compare the results of DL models, Houdini wind simulation and ANSYS Fluent (a popular and professional CFD software). The environment settings in ANSYS are the same as in Houdini.

4 Results and Discussion

Figure 4 shows the comparison of results generated from two models respectively and the real wind simulation from Houdini and ANSYS fluent. Figure 5 shows the performance comparison in training loss and FID value of the pix2pix model and the cycleGAN. In Fig. 4, the results indicate that both cycleGAN and pix2pix can perform well in generation images, which are very close to the output dataset. And FFD can predict wind environment with reasonable accuracy. The flow direction is similar between FFD and CFD, however, discrepancies are present in flow speed.

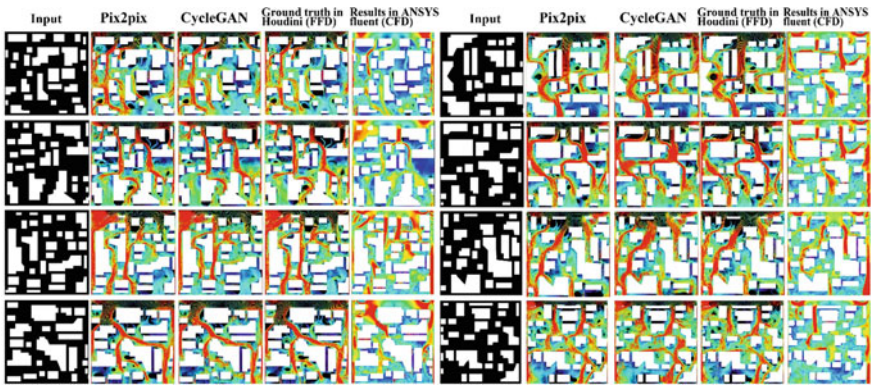


Fig. 4. Testing results comparison of two models and ground truth simulation

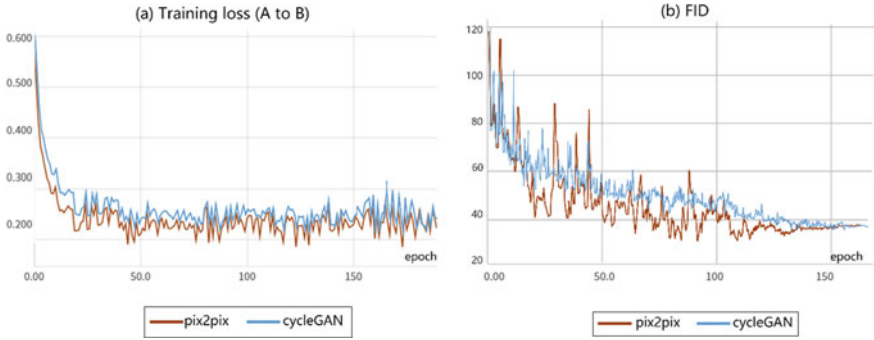


Fig. 5. Training loss diagram and FID of pix2pix and cycleGAN model

In the training loss diagram of Fig. 5, we can see that at the beginning of the training process, the accuracy and stability of cycle GAN are not as good as that of the pix2pix model, but after epoch 50, the two loss of models decreases and gradually converges. In the FID diagram, both values decline from 120 to lower than 20 after epoch 100, meaning that both of them can generate higher-quality images, the cycleGAN shows a steady downward trend, while the fluctuation of pix2pix is relatively larger. From the perspective of training time, for each training epoch, pix2pix only takes the 90–100 s on average, while cycleGAN takes 300 s on average. As a result, cycleGAN can generate images as high-quality as the pix2pix model, although it requires more training time, it saves a lot of time performing simulation to prepare paired datasets.

However, this study still has some limitations. Although DL prediction results are similar to CFD results in wind direction and speed, which can be used for preliminary design decisions, wind speed results are still not as accurate as CFD results in details. Future research could aim at using CFD samples to train a DL model for more convincing methods. The above methods are the results of software simulation. Wind simulation of the real environment can help us to correct a more accurate DL wind simulation workflow. Moreover, cycleGAN can only be used for preliminary and simple wind simulation judgment at present. The application at different scales has not been tested. Further research will utilize a larger amount of data to evaluate whether its accuracy is related to the amount of data that achieves better performance than the pix2pix model.

5 Conclusions

In this study, with the comparison of the results from the cycleGAN model and pix2pix model, we discover that cycleGAN can predict wind speed as accurate as pix2pix and it has a relatively stable state in the training process but only requires more training time. Besides, from the perspective of dataset preparation, it can greatly save time because it enables designers to input unpaired datasets of city block and wind simulation images, meaning that they don't need to perform complex and repetitive wind simulations. This study offers new thinking that enables designers to choose a faster and more accurate workflow in deep learning for the wind environment.

References

1. Toparlar Y, Blocken B, Maiheu B, Van Heijst GJF (2017) A review on the CFD analysis of urban microclimate. *Renew Sustain Energy Rev* 80:1613–1640
2. Milne-Thomson LM (1973) *Theoretical aerodynamics*. Courier Corporation
3. Blocken B, Janssen WD, van Hooff T (2012) CFD simulation for pedestrian wind comfort and wind safety in urban areas: general decision framework and case study for the Eindhoven University Campus. *Environ Model Softw* 30:15–34
4. Stam J (1999). Stable fluids. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pp 121–128. (1999, July)
5. Mokhtar S, Beveridge M, Cao Y, Drori I (2021) Pedestrian wind factor estimation in complex urban environments. In: *Asian conference on machine learning*, pp 486–501. PMLR. (2021, Nov)
6. Waibel C, Bystricky L, Kubilay A, Evins R, Carmeliet J (2017) Validation of Grasshopper-based fast fluid dynamics for air flow around buildings in early design stage. *Building Simulation*, pp 7–9
7. Calzolari G, Liu W (2021) Deep learning to replace, improve, or aid CFD analysis in built environment applications: a review. *Build Environ* 206:108315
8. Kim B, Lee DE, Preethaa KS, Hu G, Natarajan Y, Kwok KCS (2021) Predicting wind flow around buildings using deep learning. *J Wind Eng Ind Aerodyn* 219:104820
9. Brownlee J (2020) A gentle introduction to cyclegan for image translation. *Machine Learning Mastery*.
10. Kutz JN (2017) Deep learning in fluid dynamics. *J Fluid Mech* 814:1–4
11. Jin X, Cheng P, Chen WL, Li H (2018) Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder. *Phys Fluids* 30(4):047105
12. Guo Y, Liu Y, Oerlemans A, Lao S, Wu S, Lew MS (2016) Deep learning for visual understanding: a review. *Neurocomputing* 187:27–48
13. Farimani AB, Gomes J, Pande VS (2017) Deep learning the physics of transport phenomena. [arXiv:1709.02432](https://arxiv.org/abs/1709.02432)
14. He Y, Liu XH, Zhang HL, Zheng W, Zhao FY, Schnabel MA, Mei Y (2021) Hybrid framework for rapid evaluation of wind environment around buildings through parametric design, CFD simulation, image processing and machine learning. *Sustain Cities Soc* 73:103092
15. Mokhtar S, Sojka A, Davila CC (2020) Conditional generative adversarial networks for pedestrian wind flow approximation. In: *Proceedings of the 11th annual symposium on simulation for architecture and urban design*, pp 1–8. (2020, May)
16. Delua J (2021) Supervised vs. unsupervised learning: what's the difference. *Artif Intell Retrieved* 5(09)
17. Zhu JY, Park T, Isola P, Efros AA (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE international conference on computer vision*, pp 2223–2232
18. Sidefx.com. (2022). PDG|SideFX. <https://www.sidefx.com/products/pdg/>. Accessed 12 Mar 2022
19. Kaushik V, Janssen P (2015) *Urban Windflow: Investigating the use of animation software for simulating windflow around buildings*
20. Nodado CD, Yogiama C, Tracy K (2021) Towards wind-induced architectural systematization-demonstrating the collective behaviour of urban blocks as a design asset
21. Carneiro T, Da Nóbrega RVM, Nepomuceno T, Bian GB, De Albuquerque VHC, Reboucas Filho PP (2018) Performance analysis of google colab as a tool for accelerating deep learning applications. *IEEE Access* 6:61677–61685

22. Isola P, Zhu JY, Zhou T, Efros AA (2017) Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1125–1134
23. Heusel M, Ramsauer H, Unterthiner T, Nessler B, Hochreiter S (2017) Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Adv Neural Inf Process Syst* 30
24. Dowson DC, Landau B (1982) The Fréchet distance between multivariate normal distributions. *J Multivar Anal* 12(3):450–455

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

