URBAN SCALE 3 DIMENSIONAL CFD APPROXIMATION BASED ON DEEP LEARNING

A quick air flow prediction for volume study in architecture early design stage

YAHAN XIAO[1], AKITO HOTTA[2], TAKAAKI FUJI[3], NAOTO KIKUZATO[4] and KENSUKE HOTTA[5]
[1,2,5]Pollc.
[3,4]Mitsubishi Jisho Sekkei Inc.
[1]1023856216@qq.com, 0000-0002-5520-1163
[2]hottaakito@p-o.co.jp, 0000-0001-7000-243X
[3]takaaki@ty-fuji.info, 0000-0002-9182-2071
[4]nao.kikuzato@gmail.com, 0000-0001-5627-0905
[5]hotakensuke@p-o.co.jp, 0000-0003-3507-570X

Abstract. The CFD generated by an object and its surroundings is critical during architectural design. The most common method of CFD calculation is to discretize the spatial region into small cells to form a three-dimensional grid or grid point and then apply a suitable algorithm to solve the equation iteratively until the steady state, which usually takes a significant amount of time before it converges to the exact solution of the problem. Deep learning is a subset of a Machine Learning algorithm that uses multiple layers of neural networks to perform in processing data and computations on a large amount of data. This paper presents a deep learning model CNN architecture to provide a quick and approximated 3-dimensional solution for the CFD. Our network speeds up 45 times compared to the standard CFD solver. Moreover, our network is able to predict a CFD in which the wind inlet and outlet appear at the same surface of a wind tunnel.

Keywords. Urban Microclimate; Machine Learning; 3D Unet; Residual Block; 3 Dimensional CFD Prediction; SDG 11.

1. Introduction

In architectural design, the interaction between a building and its surroundings is critical to a designer's decision-making. With the development of a variety of analytical tools to assist architectural design, architects can easily complete the analysis of various indicators such as light, wind, and illuminance without mastering professional knowledge. However, when there are multiple design options, a time-consuming simulation would be unable to be applied to each one of them. Of these analyses, airflow analysis also known as computational fluid dynamics is the most expensive one. Even so, many decisions need to be based on it, such as the shape design of super

POST-CARBON, Proceedings of the 27th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA) 2022, Volume 1, 303-312.

high-rise buildings without bringing extra wind pressure to its surrounding buildings and creating a safe living environment for pedestrian and citizens. Therefore, a quick and accurate CFD simulation tool is needed. Since the middle of 2010, cloud environment simulation services have been launched one after another, and the speed of access to simulation results as well as the accuracy of the results have been attracting attention. There is a similar trend in Asia-Oceania, and the methods introduced in this paper will contribute to the development of the competitiveness of the services as well as the tools themselves.

Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical analysis and data structures to analyse and slice problems that involve fluid flows. The most common method of CFD calculation is to discretize the spatial region into small cells to form a three-dimensional grid or grid point and then apply a suitable algorithm to solve the equation iteratively until the steady state (Brunton et al., 2021). Since the number of iterations of the calculation cannot be specified, the computation often takes a considerable amount of time to converge to an exact solution to the problem.

Deep learning is an algorithm in machine learning based on characterization learning of data, which allows a system to automatically discover the representations needed for feature detection or classification from raw data. It uses mathematics to explain the relationship between data and applies the logic to new data to predict. The successful application of deep learning in computer vision, speech recognition, natural language processing, and biological information also proves its efficiency and potentiality to other applications. For example, a stable CFD pattern shows periodicity, which means the result at the same time in different periods should be the same. Then the results can be regarded as only related to the situation of objects, such as quantity, position and shape. Hence, we apply deep learning to map geometry and velocity fields efficiently and enable quick CFD prediction.

In this paper, the factors which impact CFD like seasons, temperate, humidity, sun radiation, site, and building materials are not taken into consideration.

2. Literature Review

In recent years, the booming of convolutional neural networks research on computer vision and the applications like self-driving, body temperature detection has proven that it is successful in learning geometry recognition and representation. Meanwhile, many researchers (Ahmed et al., 2020) are trying to utilize neural networks to predict CFD since it is very dependent on the geometry's condition.

Researchers (Guo et al., 2016) approximated solutions for accelerating the results with a low error rate to show how neural networks can predict the velocity magnitude field in steady-state flows. While other works (Lui and Wolf, 2019), like aerofoil design optimization and acceleration of sparse linear system solutions, also provide further contributions of neural networks' application on fluid prediction in 3 dimensions. However, in these studies, although the prediction of the CFD on 2-dimensional can get a high degree of reduction in a short time, the default condition is that the shape of the other section of the objects is consistent with the shape of the observation plane or remains unchanged. Meanwhile, most studies on predicting CFD on 3-dimensional only considered a single object, and the wind direction is fixed. These do not correspond to reality.

In researcher MD Ribeiro's work, the U-Net network has been proven to be capable of CFD prediction in 2D version. Therefore, we propose an approximation model based on the general U-Net structure to predict the 3-dimensional velocity field of multiple objects for conceptual study at the early stage. This paper visualizes the simulation results digitally during the conceptual design process, which enable more reasonable decision-making and then achieve sustainable cities.

3. Methodology

The approach is combined by 5 steps: 1) procedural geometry preparation; 2) CFD results collection; 3) dataset preparation; 4) ANN (Artificial Neural Network) training; 5) testing and evaluation.

3.1. TRAINING DATASET PREPARATION

A city has many buildings that are interrelated and have a complex appearance. In order to approximate the actual situation, CFD under the mutual influence of multiple different objects will be the primary data source of this research. 20 random objects in different heights, shapes and scales inside a bounding box of 192*192*96 meters are generated by grasshopper to simulate city buildings. Buildings' height varies from 10 meters to 80 meters. Moreover, the grid size for CFD calculation is 1 meter. (Figure 1)

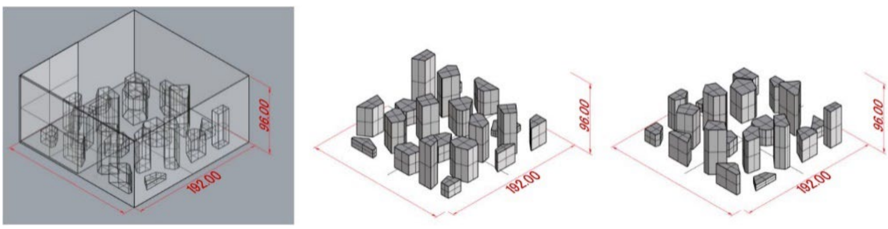

Figure 1. Wind tunnel scale (left) and other two cases

The representation of geometry needs to reflect the relative relationship between the building's shape and wind tunnel and the scale of the wind tunnel. So, we choose SDF (Signed Distance Function) (Guo et al., 2016) (Figure2 left), the shortest distance from a space point to the nearest surrounding objects, to reflect the relative position of the object and the wind tunnel. And we take reference from DeepCFD (Ahmed et al., 2020) and create WSD (non-slip wall side distance) value, distance from the wall of the wind tunnel to the central axis of the ground in the direction of the wind (Figure2 right). To make the neural network more accurately identify whether a point is inside of the objects, the SDF value of an inside point is set to -50. Both SDF and WSD were calculated by grasshopper, and for each case, SDF values remain the same regardless of the wind direction.

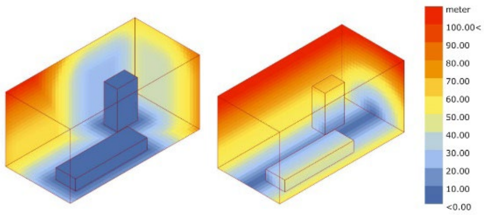

Figure 2. (left) Signed Distance Function in 3D domain; (right) non-slip Wall Side Distance in 3D domain

3.2. CFD SIMULATION

CFD result is solved by AKL FlowDesigner, a computational fluid dynamics simulation software that supports models from Rhinoceros and is able to export the velocity result on grid points. The generated objects are regarded as obstacles without any material, so the heat absorption and emission would not impact the wind speed and direction. Each case is simulated with the wind in 4 directions: south, east, north and west. The actual wind is different from the wind tunnel set in the experiment with only one inlet and outlet. It is affected by objects' location, shape and quantity that multiple inlets and outlets could exist on one plane. Considering this, a CFD result with a smaller scale of 96*96*96 meters which is taken from the overall results of each case will contain the feature. Twenty-five small boxes data are taken from one CFD simulation, where 75% of each small box is overlapped with other boxes (Figure3 left), which allows the network to thoroughly learn the relationship between the different boundary wind directions and the generated wind fields.

In this paper, we use LGR (latent geometry representation) (Ribeiro, 2020) to represent the flow region (0 for the obstacle inside condition, 1 for the flow region, 2 for the upper and surrounding no-slip wall condition, 3 for velocity inlet condition, 4 for velocity outlet condition, 5 for obstacle' surface condition and 6 for ground condition). For the LGR value of a point on a small-scale box's surface, the inlet and outlet condition are determined by the product of its wind direction and the average vector of the surfaces (Figure3 right).

Three hundred small boxes with a size of 96*96*96 meters' geometry data and its corresponding CFD data are prepared as the dataset of our neural network.

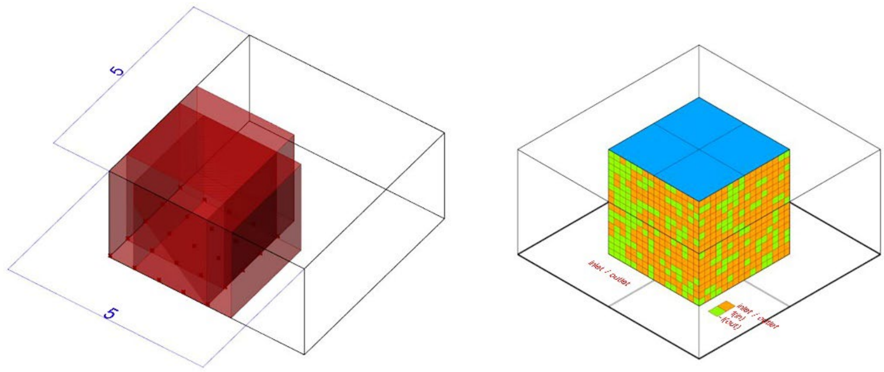

Figure 3. (Left)small-scale training dataset preparation; (Right) different LGR value: 1. inlet: orange; 2. outlet: green; 3. non-slip wall: blue; 4. Ground: brown

3.3. NETWORK INTRODUCTION

An autoencoder (Figure4), a type of artificial neural network, learns a representation of a set of data and has two main parts: an encoder that maps the input into the code and a decoder that maps the code to a reconstruction of the input. The autoencoder network can discover a coordinate and a decoder to obtain a latent space containing minimal information to describe the whole system. So, it suits the great dimensional problem such as a fluid flow which would require billions of degrees of freedom to represent but only a few of them are essential and have low dimensional patterns.

The U-net network is similar to the autoencoder model, consisting of an encoder and decoder. The difference is that U-net uses a concatenation layer to connect the encoder and decoder, which recovers information loss in the down-sampling process and gives localization information to let the output back to the exact size of the input. Other researchers have proven that the U-net network is powerful enough for CFD and has achieved high-speed and accurate prediction on one single 2D object and one single 3D object (Guo and Li, 2016; Ribeiro et al., 2020). Since in reality, the CFD impacted by the interaction among buildings is more complicated to learn. To predict a result close to the real situation, a more generalized and more robust neural network is necessary. Then the structure of our network is based on the existing structure of 3D U-net, making every path from the network's input to its output is composed of a residual sub-network. The residual block can increase the depth of the network without losing accuracy.

We implemented our network based on Lee K's work, in which a 3D Unet network with residual block has been successfully applied to brain image segmentation tasks (Liu et al., 2020). The basic building block of our network consists of a single convolutional layer followed by the residual block, and a summation join replaces the concatenation joining. Our network-building block’ s order is group normalization layer, convolutional, and Relu activation function. Upsampling is implemented with stridden transposed convolution and downsampling with max pooling (Figure 7).

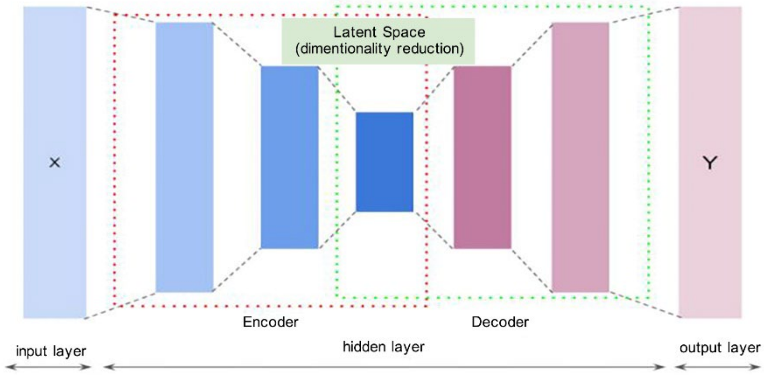

Figure 4. Autoencoder structure

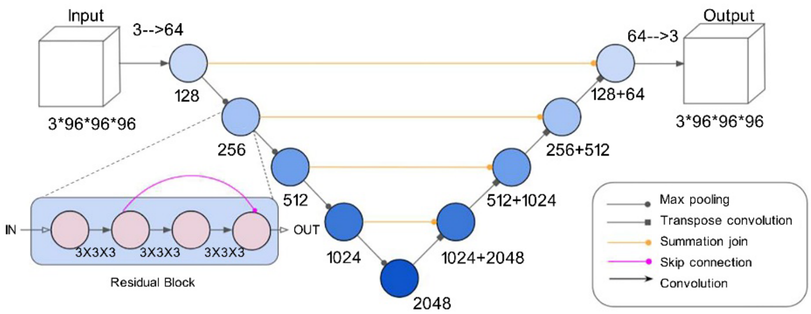

Figure 5. network structure

3.4. TRAINING

We implemented the network using Pytorch. Regarding the learning process, the Adam optimizer was employed with batch size, and weight decay was set to 0.05. Three hundred datasets are divided into training sets and validation sets at 80% to 20%. At the same time, the loss function is the mean squared error (squared L2 norm) which measures the output velocity in Ux, Uy and Uz directions. The initial learning rate is set to 1e-5 and is gradually decreased by a learning rate scheduler which allows dynamic learning rate reduction based on the validation measurements. The training time for one epoch is approximately 6500 seconds.

4. Results and Performance Analysis

To evaluate the accuracy of the network, 50 cases were tested, and the velocity field prediction is compared with the generated actual value from FlowDesigner. We visualize one prediction and true value in Figure 6 to understand the model's accuracy. The performance of our network is evaluated from three cross sections of the velocity fields on the X-Y, X-Z and Y-Z planes and shows the magnitude of the difference in wind's three directions in Figure 7.

Test wind tunnel size is the same as the training sets. However, considering the CFD solver only uses CPU for calculating, we compared our network's time-consuming prediction and simulation by Flow Designer on CPU. The average simulation time for the current wind tunnel is 3 minutes, and the prediction time is 4 seconds, which means our network has speed up 45 times.

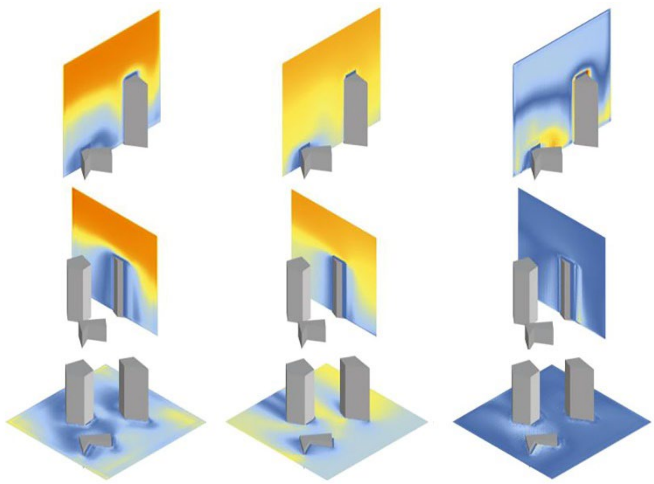

Figure 6. (left side images) CFD prediction(0-10m/s); (middle images) true value(0-10m/s); (right side images) differential(0-5m/s)

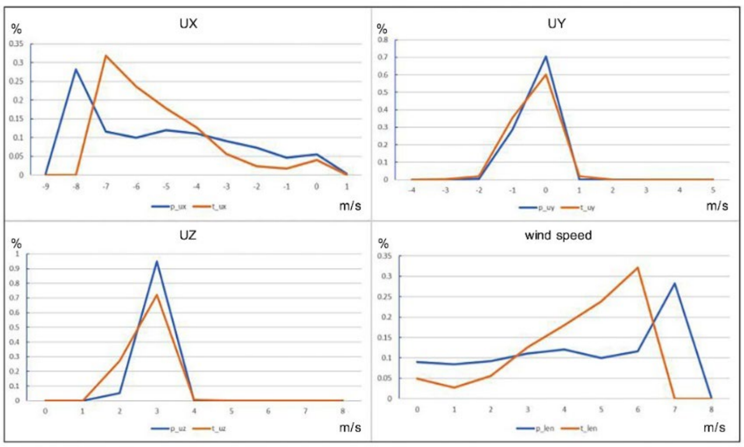

Figure 7. (horizontal axis: speed (m/s); vertical axis: distribution (%))The difference between the predicted value and the true value of the three wind directions and wind speed (blue: prediction; orange: true value)

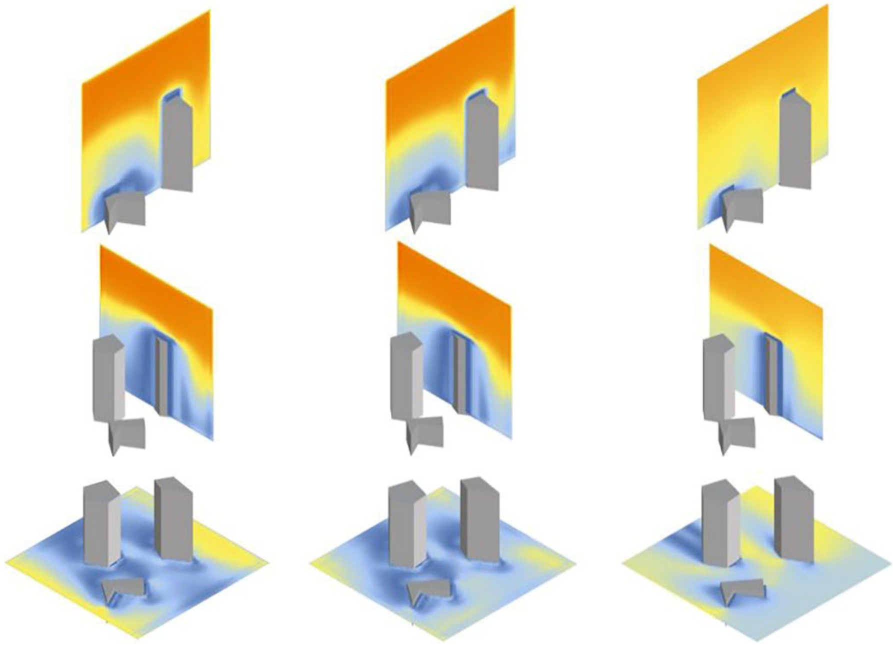

Figure 8. (left side images) prediction with actual boundary condition(0-10m/s); (middle images) previous prediction(0-10m/s); (right side images) true value

Figure 6 predicts that the boundary conditions are set as non-slip walls, two surfaces and their positions are parallel to the wind direction; the ground; the air inlet, one surface which is perpendicular to the wind direction and let the wind get in; the air outlet, one surface is positioned perpendicular to the wind direction and as the main exit; and the surface parallels to the ground. The prediction in Figure 6 shows that the network can learn low dimensional latent space like the variation law of wind speed along the height, the changing law of wind speed, and wind direction after passing through an object from the current data. However, the velocity field's feature has not been well-learned yet, for example, the changed wind direction and speed at objects' surfaces.

There is more than one wind condition on the surface of the wind tunnel. In other words, there are both air inlets and outlets on the same surface. Therefore, we predict the same objects with the acceptable boundary conditions extracted from FlowDesigner when preparing the actual value. As a result, the overall difference between the new prediction (the left image in Figure 8) and the actual value is reduced by 10% compared to the previous prediction, which shows that our network can provide more accurate results under the proper boundary conditions.

Usually, for different designs, the surrounding objects that impact the CFD simulation of the target area need to be taken into calculation every time. The more design options there are, the greater the total CFD calculation times. After obtaining the conditions of the boundary which has the negligible impact to the target area, our current network can be applied to the partial design of large buildings or the volume study of small buildings.

5. Future Work and Conclusion

In this paper, we proposed an efficient and quick way for approximating a city-scale CFD in 3 dimensions which enables architects to get the velocity field of their design during the volume study. In addition, this article is the starting point for our future development of the grasshopper plugin, which provides instant feedback on CFD.

The basic equation describing the characteristics of fluid motion is the Navier-Stokes equation, a nonlinear differential equation, which expresses the relationship between fluid motion and the force acting on the fluid. It contains the fluid velocity, pressure, density, viscosity, temperature, and other variables closely related to the space position and time. Moreover, the computational difficulty of 3-dimensional is exponential times that of 2-dimensional, which means in order to predict a 3-dimensional CFD accurately, the network needs to be deep enough. We first tried with four encoders without applying residual blocks during the network construction. The prediction shows that objects' edges can be recognized, whereas the repeating pattern of swirling vortices behind the object is hard to be well predicted. After we added residual blocks and increased the depth of our network to six, the previous problem was solved. Nevertheless, compared with the ground truth, the wind direction and wind speed changed by the pressure field, which is impacted by objects' number, relative position and boundary situation before the wind reaches an object, has not been well learned yet. What is more, our current predictable scale is 96*96*96, which is not enough for an accurate urban scale CFD prediction.

For our future work, we intend to enlarge our prediction scale and expand the dataset in terms of the more complex shape of objects and flow conditions, so that the prediction can be applied to the actual work. We have proved that 3D U-net with residual blocks is capable of 3-dimensional CFD prediction issues, but the number of the network's trainable parameters will increase sharply as the network deepens. Therefore, we intend to reconstruct the current network by combining the autoencoder and regression model. After the low-dimensional latent space is found by the autoencoder, the regression model will be built inside of it (Brunton et al., 2021).

Acknowledgements

This work was funded by Mitsubishi Jisho Sekkei Inc.

References

Ahmed, S., Dengel, A., Rehman, A., & Riberio, M. D. (2020, April 19). *DeepCFD: Efficient Steady-State Laminar Flow Approximation with Deep Convolutional Neural Networks*. arXiv Physics. Retrieved September 7, 2021, from https://arxiv.org/abs/2004.08826.

An, W., Liu, X., Lyu, H., & Wu, H. (2021). A generative deep learning framework for airfoil flow field prediction with sparse data. *Chinese Journal of Aeronautics*, 35(1), 470-484. https://www.sciencedirect.com/science/article/pii/S1000936121000728.

Bhat, B., Huval, B., Manning, C. D., Ng, A. Y., & Socher, R. (2012). Convolutional-Recursive Deep Learning for 3D Object Classification. In *Advances in Neural Information Processing Systems, 25 (NIPS 2012).* The Conference on Neural Information Processing Systems.

Brunton, S. L., Callaham, J. L., & Loiseau, J. C. (2021). *On the role of nonlinear correlations in reduced-order modeling*. arXiv Physics. Retrieved September 7, 2021, from https://arxiv.org/abs/2106.02409.

Chen, S. F., Hsiung, P. A., & Utomo, D. (2017). Landslide Prediction with Model Switching. *Applied Sciences,* 9(9). https://www.mdpi.com/2076-3417/9/9/1839.

Davila, C. C., Mokhtar, S., & Sojika, A. (2020). Conditional Generative Adversarial Networks for Pedestrian Wind Flow Approximation. *The 11th annual Symposium on Simulation for Architecture and Urban Design, SimAUD2020* (pp. 469-476). The Symposium on Simulation for Architecture and Urban Design (SimAUD).

Ding, C., & Lam, K, P. (2019). Data-driven model for cross ventilation potential in high-density cities based on coupled CFD simulation and machine learning. *Building and Environment,* 165, Article 106394. https://www.sciencedirect.com/science/article/abs/pii/S0360132319306043.

Guo, X., & Li, W. (2016). Convolutional Neural Networks for Steady Flow Approximation. In *KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 481-490). The annual ACM SIGKDD conference.

He, Y., Liu, X. H., Mei, Y., Schnabel, M. A., Zhang, H, L., Zhao, F. Y., & Zheng, W. (2021). Hybrid framework for rapid evaluation of wind environment around buildings through parametric design, CFD simulation, image processing and machine learning. *Sustainable Cities and Society,* 73, Article 103092. https://www.sciencedirect.com/science/article/abs/pii/S2210670721003759.

Huang, Y., Lu, X., Sun, C., Zhang, F., Zhao, P., & Zhao, X. (2021). Automated Simulation Framework for Urban Wind Environments Based on Aerial Point Clouds and Deep Learning. *Remote Sensing,* 13(12), 2383#. https://www.mdpi.com/2072-4292/13/12/2383#.

Jain, V., Lee, K., Li, P., Seung, H. S., & Zung, J. (2017). *Superhuman Accuracy on the SNEMI3D Connectomics Challenge*. arXiv Computer Science. Retrieved September 7, 2021, from https://arxiv.org/abs/1706.00120.

Liu, Ping & Dou, Qi & Wang, Qiong & Heng, Pheng-Ann. (2020). An Encoder-Decoder Neural Network With 3D Squeeze-and-Excitation and Deep Supervision for Brain Tumor Segmentation. *IEEE Access,* 8, 34029-34037. https://ieeexplore.ieee.org/document/8998244.

Lui, H. F. S., & Wolf, W. R. (2019). *Construction of reduced-order models for fluid flows using deep feedforward neural networks.* arXiv Physics. Retrieved September 7, 2021, from https://www.cambridge.org/core/journals/journal-of-fluid-mechanics/article/abs/construction-of-reducedorder-models-for-fluid-flows-using-deep-feedforward-neural-networks/ECEC52E32AEBBEA049CF26D6C79EE394.