Generative Machine Learning Models for Airflow Prediction of Architectural Spaces

FERNANDO J. CLAUDIO RODRÍGUEZ
University of North Carolina at Charlotte

Keywords: Machine Learning, Artificial Inteligence, Natural Ventilation, Computer Fluid Dynamics (CFD) & Architecture

Buildings have been identified as one of the biggest contributors of negative environmental impacts worldwide, more specifically energy usage due to the use of air conditioning and mechanical ventilation. Strategies such as cross-ventilation have become a reliable alternative to diminish some of these effects. However, designing for cross ventilation is no easy feat, as it requires architects and designers to study in detail the building context, overall massing design and building enclosure to maximize airflow potential. On the other hand, Computer Fluid Dynamics (CFD) airflow simulations are not used as often in architectural settings primarily due to time constraints, lack of performance metrics and quality assurance. The proper use of CFD airflow simulations involves a complex setup and run-time process, due to the large mathematical calculations involved.

This study aims to apply existing generative machine learning algorithms to compute CFD wind velocity simulations to significantly shorter run times while maintaining a relatively high accuracy level, during the initial design stages. To test the proposed hypothesis, multiple machine learning models were created, trained, and tested. The evaluation metrics for these models consisted of using different image similarity methods to compare the images produced by the machine learning model to their CFD engine counterparts.

The results obtained indicate that GAN application for CFD airflow predictions can produce acceptable results showing a significant run time difference of over a minute between the CFD simulation and the machine learning model. Having evaluated and proven this study as a proof of concept, this can set the precedents for further research on the use of CFD airflow simulations and machine learning within architectural practice. Allowing architects and designers to incorporate the use of CFD airflow simulations within their workflows.

INTRODUCTION

Over the past couple of decades, buildings have been identified as contributing to the negative environmental impacts worldwide [1]. Being responsible for 40% of carbon emissions [2], with 56% of all energy use going to air conditioning and mechanical ventilation in hot-humid climates [3.] As energy consumption increases, passive cooling strategies, such as cross-ventilation, have become a reliable alternative. However, designing for cross ventilation is no easy feat, as it requires architects and designers to study in detail the building context, overall massing design and building enclosure to maximize airflow potential.

Technological advances, on the other hand, have simplified and reworked the way architects design, including the use of environmental analysis. Using different software, these processes can be performed from a single two or three-dimensional model inside a virtual environment. However, as many architects have identified: "***design time is usually quite short and anything adding to that is an obstacle.***" [4], in addition to the lack of quality assurance and performance guidelines. It is uncommon for architects and designers to incorporate these testing simulations within their workflow.

Due to improvements in the creations of user-friendly tools, architecture firms are increasingly implementing daylight, radiation, glare, and energy simulations into their workflows. These tools rely on efficient, straightforward, and well-documented engines that produce data graphically and legibly. Natural ventilation studies, on the other hand, are much more complicated due to exponentially increasing mathematical calculations required to simulate the interaction between airflow and the defined enclosure.

As data has revolutionized how we use information, new methods are being developed to manage the increasing size of data sets. A solution to automize the data management process has been the incorporation of artificial intelligence, specifically, the use of machine learning algorithms. The reliance on these algorithms is significant as computers can now learn from these increasing data sets and perform multiple functions from

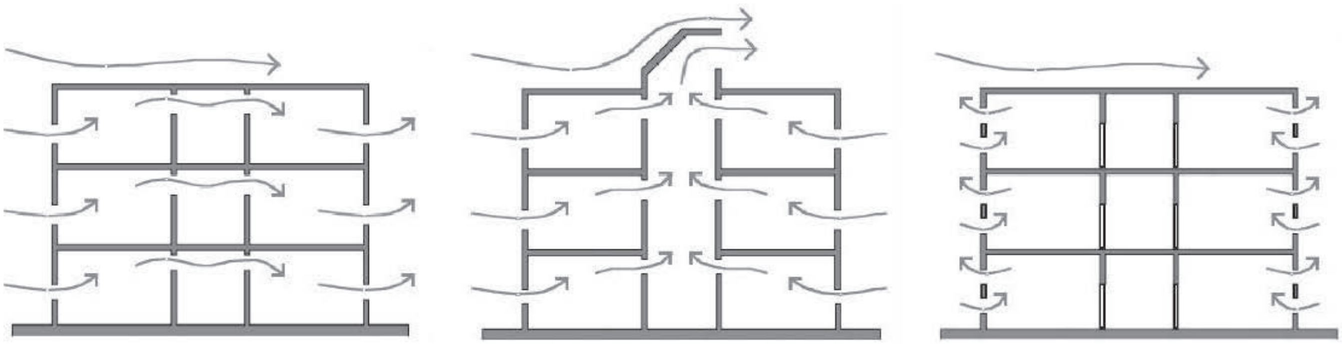

Figure 1. Natural Ventilation Strategies: (Left) Cross Ventilation, (Middle) Stacked Ventilation, (Right) Single Sided Ventilation
T. Stathopoulos, "Wind and Comfort," Jul. 2009.

complex mathematical calculations to image generation, with minimal human intervention.

This study aims to apply existing generative machine learning algorithms to compute Computer Fluid Dynamics (CFD) wind velocity simulations to significantly shorter run times while maintaining a relatively high accuracy level. These generative algorithms aim to provide faster run times for CFD simulations than traditional simulation processes and obtain relatively accurate results. All with the intent to promote the use of cross ventilation and the incorporation of CFD models within design workflows.

BACKGROUND

As diverse topics from architecture and machine learning will be covered throughout this study, this section explains a general overview of these, ranging from natural ventilation implementation and basic cross ventilation strategies to basic concepts of the functionality of machine learning algorithms.

NATURAL VENTILATION & COMPUTER FLUID DYNAMICS:

Natural ventilation consists of moving air from an external source to an indoor space due to the changes in pressure without any mechanical system. Its usage has many purposes, from air quality control to passively cooling an interior space. Three predominant approaches include the following: cross, stacked, and single sided ventilation[5]. On the other hand, CFD is a combination of multiple fields to simulate the movement and reaction of various fluids. In architecture, these are used to examine natural ventilation, infiltration, and dispersion of air contaminants. The simulation process consists of generating a geometric model and divide it into a cell grid [6]. After inputting a set of criteria, such as wind velocity and temperature, the simulation will calculate how the desired airflow is affected by the input geometry.

Two approaches when simulating airflow are coupled and decoupled. The first involves connecting the outdoor and indoor airflow in a single model. The counterpart only involves analyzing either indoor or outdoor environments [7]. As studies suggest ([8,9,10]), the use of the coupled models is the optimal approach when it comes to cross-ventilation studies, as it considers how the pressure of the building envelope will be affected by the presence of the opening.

MACHINE LEARNING:

Machine learning is a subset of artificial intelligence, which has the quest to teach computers to perform a series of tasks without being explicitly programmed for them. This process is comprised of three key components. First, the data set from which the model will learn during its training process, the format of these can range from all kinds of data types. Secondary, are the features telling the machine the essential factors to be aware of. Lastly is the algorithm, which provides a particular method to solve the problem [11]. As there are various kinds of algorithms, there are also multiple ways to teach or use a machine to help solve problems. As described in Python Machine Learning by Raschka & Mirjalili, this consists of three methods, supervised, unsupervised, and reinforcement learning. The supervised machine learning model is trained with a dataset containing labeled information. Its outcomes are known, allowing the model to receive new unlabeled data as input to make predictions. In reinforcement learning, the model generates a "reward signal," which is not the correct answer, but a comparison of how well the performed action relates to the reward function. Lastly, in unsupervised learning it is up to the machine to use specific techniques to find patterns and relationships within the data [12].

GENERATIVE ADVERSARIAL NETWORKS (GAN):

Generative Adversarial Networks (GAN) uses generative modeling in combination with convolutional neural networks. The final purpose of GAN is to generate new data predictions from nothing. The network is composed of two models, the generator, and the discriminator. The first generates new examples, while the second tries to identify whether the generator's examples are real or fake in a "zero-sum game" [13]. This kind of model's primary data type are images, although other types have been used. Both networks are trained in an alternating

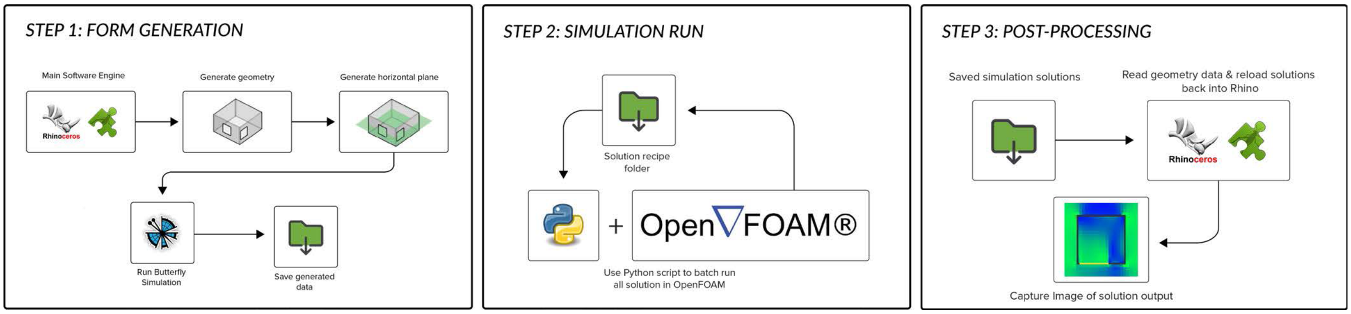

Figure 2. Data Set Creation Steps (1. Form Generation (Left) | 2. Simulation Run (Middle) | 3. Post Processing (Right))

state during the training process until the discriminator model cannot tell the difference between real and fake images created by the generator [14].

METHODOLOGY & DATA SET CREATION:

As an approach to the problems previously stipulated, this section will explore the use of supervised machine learning algorithms in combination with coupled airflow CFD analysis in architectural buildings. The process consists of three major steps: creating a data set, training the machine learning model, and evaluating its results. For this study, only cross ventilation was studied on a two-dimensional plane.

The creation of the data set consists of three stages: form generation, simulation run, and post-processing. The form generation step was made inside Rhinoceros's Grasshopper, beginning with the generation of a 3.3m x 3.3m square for the base plan, offsetting 0.1m walls on all sides and extruding 2m in height. After the generation of the walls, panoramic windows were created, which were subdivided into three sections, providing twelve windows for this scenario. To limit the number of variations, combinations were limited to between two and four windows. This leads to a total of 781 variations. Another applied constant was the wind speed at 4.5m/s and its direction, coming from the south. To ensure the best run time/accuracy relationship, five grid densities were evaluated ranging between 1.0 and 0.1 meters. Based on the obtained results, 0.2 was determined to be the best alternative as it provided higher accuracy results than the coarse alternatives and close to the finer ones. Run times resulted in an average of 64 seconds per simulation for the 0.2 density, while the finer 0.1 density had an average of 206 seconds.

The second phase of the data set creation, the simulation run stage, was performed in two steps, recipe generation and simulation execution. The CFD engine used for this study was OpenFOAM, which is the most validated open-source engine for running advanced simulations. This process consisted of passing all the final geometric iterations through the Butterfly plug-in, which prepared a specific recipe for the engine to understand the required steps to run the actual CFD simulation. Then consisted of batch running all the recipes inside the engine's terminal and store the results.

The third stage was post-processing, performed inside Rhinoceros and Grasshopper. This stage consisted of reloading the original three-dimensional model and loading the butterfly recipe back into the Grasshopper. Once both were loaded, a 2048px x 2048px image was captured of the building's plan view alongside the airflow simulation results and stored in an external folder. These will be the input data for the machine learning model. In addition to the CFD map images, an image without the simulation outcome was also captured to serve as a base input.

After all the data was collected, it was split into two groups, testing, and training. This division was done at an 80/20 ratio (training, testing) to use the same dataset to train and test the model without generating an additional data set. The purpose of using an 80/20 ratio as opposed to a 60/40 was to provide the machine learning model with a larger data set to train as it increases the model's accuracy.

MACHINE LEARNING & TRAINING:

For the machine learning model, the Pix2PixHD algorithm by NVIDIA was used as a base. This framework uses the CycleGAN technique, which automatically trains the image-to-image translation models without using paired examples—instead of using a collection of images, having a source and a target, which do not need to be related. Multiple models were trained during this stage to identify how manipulating hyperparameters would affect the model's efficiency when predicting airflow patterns.

The training parameters were the following: the data set contained no labels, epoch checkpoints saved in increments of five until reaching one hundred, images were not cropped or resized, images were flipped, and loaded at a resolution of 512px by 512px. Two discriminators were used to reduce the possibility of overfitting. The main parameter altered was the batch size, evaluated in 1, 3, and 5 variations. Batch size refers to the number of images the model will use during its training in

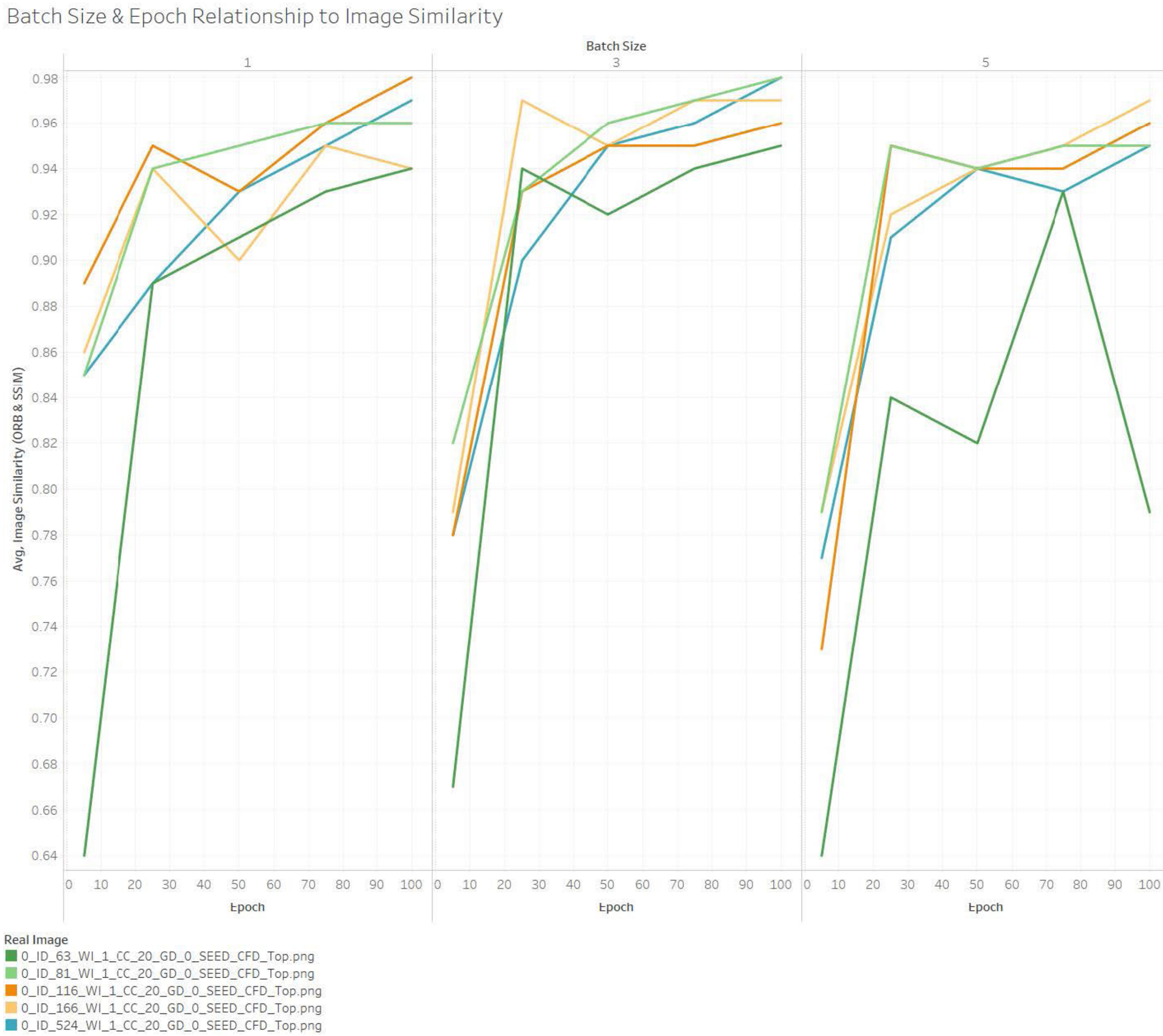

Figure 3. Epoch & Batch Size Graph in Comparison in Relation to Image Similarity

each iteration or epoch. In addition, the dataset was modified by adding additional padding to the images. After setting these parameters, the models began training with a mixture of using Google's Collab and local resources. All models were trained for one hundred epochs and took an average of eight hours to reach the 100-epoch mark.

DATA EVALUATION

As other machine learning models are trained with a loss function until convergence, evaluating GAN's results is not straightforward [15]. Given that no objective loss function is used to train the GAN generator, there are no assets to measure its development and progress. As a result, given that GAN's are image-based, the best method to determine their efficiency is upon visual inspection, however, these can be subjective.

In contemplation of this situation a methodical approach was devised, which consisted of measuring the similarity between the image produced by the GAN model and the Butterfly engine. These were compared using two techniques within a python script using the PIL and cv2 image libraries. The ORB (Oriented FAST and Rotated BRIEF) feature matching technique identifies different characteristics within the images to compare them. This method was selected compared to others as it is open source and requires less computational cost [16]. In addition, images were evaluated by their structural similarity (SSIM), which compares images on three main metrics: luminance, contrast, and structure [17,18]. This process takes a more comprehensive approach; evaluating images by looking at a group of pixels

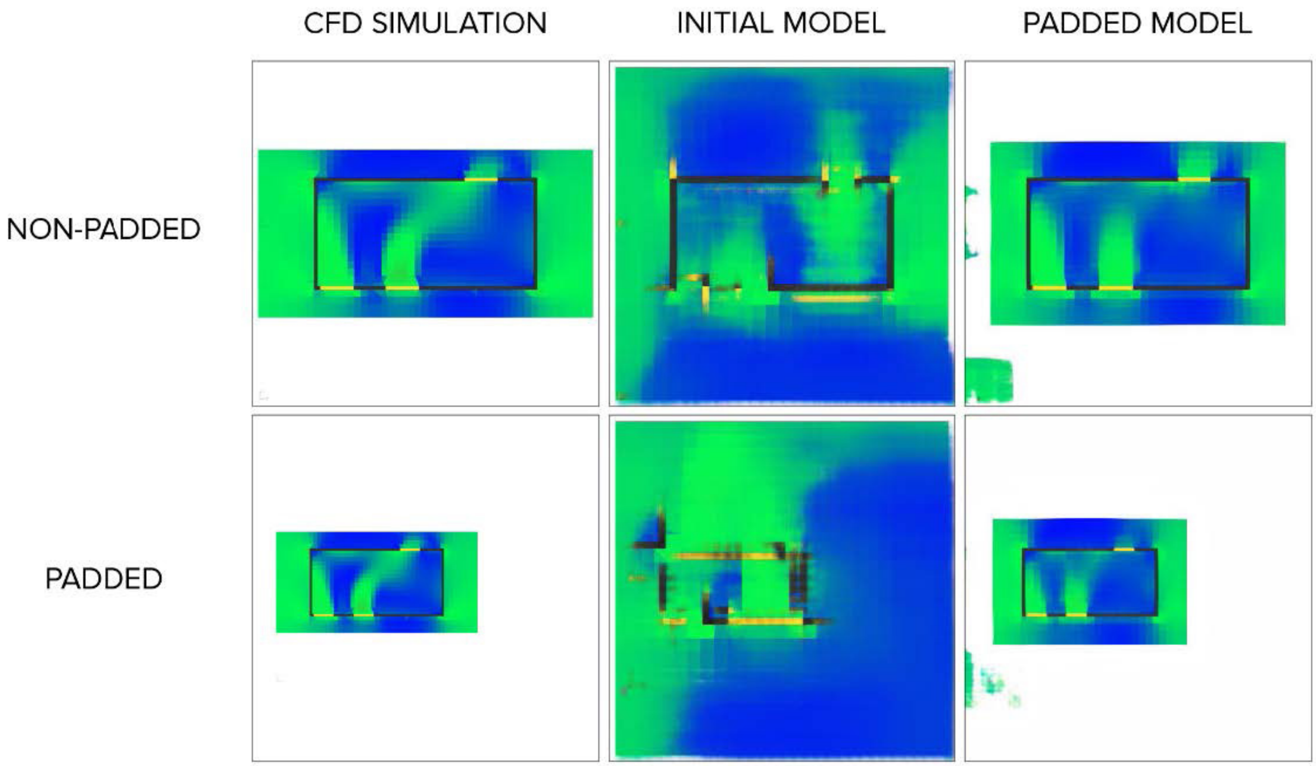

Figure 4. Test Results of Rectangular Floor Plan on Both Initial and Padded Machine Learning Model

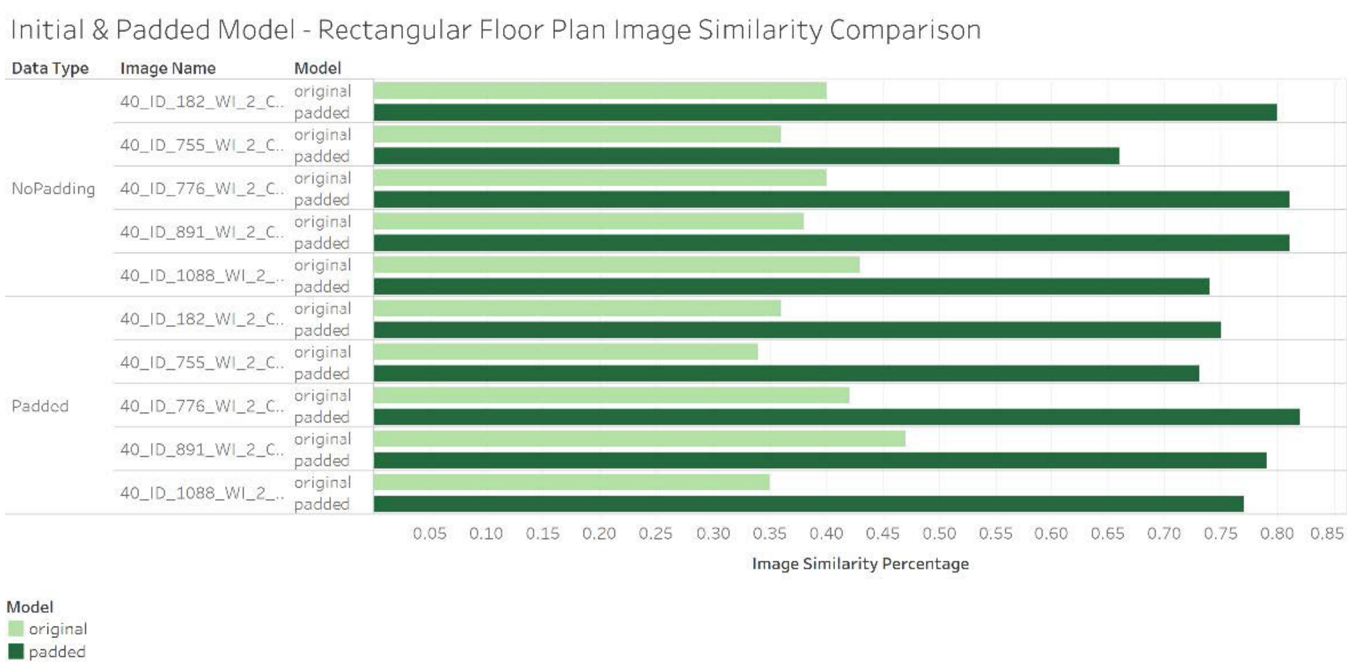

Figure 5. Initial & Padded Machine Learning Model for Rectangular Plan Image Similarity Comparison

instead of a pixel to pixel-based to determine their similarity. This provided a similar approach to how humans would perceive images. Both evaluation methods returned a number from 1 to -1, the highest indicating the most resemblance between the images and lowest indicating the least similarity between the images. As these metrics evaluate images differently, the average from both methods was calculated to obtain an objective result. These methods were applied at different training stages of each model to examine their learning process and how each of the different parameters affected its learning curve. In addition to comparing the image similarity between the image produced by the GAN model and the Butterfly engine, run times between the two models were also recorded and evaluated using a custom python script.

RESULTS AND DISCUSSION

After generating the data set, training the machine learning models, and evaluating the results, the proposed hypothesis of using GAN to obtain relatively accurate results for CFD airflow simulations was tested. Multiple models were evaluated at different training intervals to observe how their training progressed. After identifying the most effective model, this one was tested using a different floor plan layout to observe its prediction capabilities to unseen data and compare it with the initial data set.

INITIAL & PADDED DATA SET:

To evaluate the previously explained methodology, an initial machine learning model was trained using a data set of 624 images for 100 epochs. The data set consisted of images of a single square floor plan with various window configurations. After finishing the training phase, the model was tested using 15 unknown images. To the naked eye, images would appear almost identical, obtaining a structural similarity index of 95%, on average, when compared to their CFD simulation counterparts. When evaluating run times, the traditional CFD simulations took an average of 62 seconds to complete, while the machine learning model took an average of 0.2 seconds. These results present a clear indication of the efficiency of GAN to quickly generate accurate predictions. The images produced began to support the efficiency of these modes when being applied to CFD airflow simulations. However, when observing the results during the different training stages a problem was found. Within the first couple of epochs, the model produced substantially accurate results, raising questions about the efficiency of the data set or the hyperparameters used.

After analyzing these concerns, the model was presented with a set of rectangular floor plans, which produced unsatisfactory results. These showed how the model tried to generate the square floor plan in the exact location, indicating it had memorized specific pixels of the images, which caused rapid and accurate results. After further evaluation it was found this overfitting was caused as all images contained a centered floor plan, which helped the model predefine its location. Even after flipping the images, these presented no notable change in the training set, given they were symmetrical. To correct this problem, a new data set was created based on the initial one but adding padding around the existing images to change the scales of the floor plans while maintaining the same square aspect ratio. This shift helped avoid having the floor plan always centered on the image and at the same scale, providing an additional level of complexity for the model to predict.

As a next step, three different models were trained for 100 epochs using 624 images of this new data set, changing their batch size to observe how it would affect the training process of the models. The batch sizes used were 1, 3, and 5. After training all these models, they were tested at epoch 5, 25, 50, 75, and 100 epochs to observe the development of the model at different training intervals. The number 5 checkpoint was selected to compare the models' learning results with the initial data set model. For these testing, the previous two similarity methods were applied (ORB & SSIM), in addition to the recording of run times.

PADDED DATA SET RESULTS:

These machine learning models were tested using a set of five images containing the same square floor plan with different window combinations and image compositions. As a result of training the different machine learning models, visually, all models began their training producing similar results. After passing the 25-epoch mark, all three models produced images with an average of 90% similarity with the original CFD images. The model with a batch size of 3 produced the most accurate and consistent results at the end of the training phase, followed by the model with a batch size of 1, and lastly, the model with a batch size of 5. This last model appeared to have trouble predicting images even after the 100-epoch mark, as one of the tested images obtained the lowest scores of 79% similarity when compared against their CFD counterpart.

Regarding run times, it took on average 60 seconds to perform an analysis on the CFD engine, in contrast, of 0.12 seconds for the machine learning model. Except for the first image to go through the model as it took on average 1.38 seconds, given it had to generate the models and prepare the neural network. This accounts for a difference of 600 times faster than the CFD engine when presented with similar data. However, this difference is to be expected, as similar results were observed in the work of Layout 5 [19] and Kacper Radziszewski & Marta Waczyńska[20] in their daylight analysis. Nonetheless, this emphasizes the efficiency of machine learning models and supports the arguments made in this study to help designers obtain relatively accurate CFD airflow predictions in less time.

RECTANGULAR FLOOR PLAN TYPOLOGY TEST

These machine learning models will eventually be exposed to unknown data and generate predictions based on their training. Based on the results from the previous tests, the model with the best performance was the one with a batch size of 3. This model, alongside the original one trained with the initial data set, was exposed to a rectangular floor plan to test their prediction capabilities when presented with unseen data. For this test five different window combinations were selected on a rectangular floor plan to provide different scenarios. Given that the original model was not trained on images that contained padding, both were tested with a set of images with padding and without padding.

As a result, the initial machine learning model performed poorly, as it could not generate satisfactory results of the airflow pattern for the rectangular floor plans. However, it performed marginally better when presented with the non-padded images, as in this case, even though the airflow patterns are unclear, a floor plan layout was defined. The model trained on the padded images outperformed by a large margin the original model. It produced relatively accurate results of the airflow patterns on both the padded and non-padded images. When comparing the generated images with their simulation counterparts, the original model obtained an average of 39% image similarity compared to the padded model with an average of 79%, indicating a difference of 40%.

CONCLUSION & FUTURE WORK

The result obtained in this research indicates that GAN application for CFD airflow predictions can produce acceptable results for designers, allowing them to advance their design process through an alternate method with marginal time differences compared to traditional CFD simulations. Results show a significant run time difference of an average of 60 seconds between the CFD simulation and the machine learning model. The GAN models generated predictions six hundred times faster on both similar and unknown data, generating relatively accurate results with a range between 79 to 95 percent image similarity with the CFD simulation output. While seconds to a minute difference might not appear significant, minutes can grow exponentially to hours and even days when performing complex CFD simulations. Therefore, these time differences indicate a significant cut in run times.

Alternating different hyperparameters, such as adding more discriminators or increasing the batch size, might help produce more accurate results when using a larger data set. However, one of the major factors for a successful GAN model is the data set used for training. As observed in this study, the padded data set provided more varied and diverse information when compared to the initial data set—resulting in a 40% increase in image similarity to the actual CFD simulation, even when tested with unknown floor plan layouts. The variations referred to do not only account for the floor plan and window configurations themselves but also image composition—alternating parameters such as floor plan location, scale, and orientation within the image itself.

This project aims to expand the floor plan typologies to more complex forms and window configurations for future work. Alternatively, this research aims to explore other machine learning algorithms such as linear regression and other ANN and compare them alongside GAN to identify the most efficient machine learning algorithm for the desired CFD airflow simulation. The long-term goal is to convert the trained machine learning model into a Grasshopper and Dynamo Plug-In. After implementing the plug-in, user studies would be conducted with architects to observe how this system would help them within their design workflow and implement them within their practice. This study is a small starting point for the application of GAN for CFD simulations. The current goal is to provide additional insight for future researchers to continue developing the use of GAN for different CFD simulations and incorporate them into architectural workflows.

ENDNOTES

1. Naief A. Aldossary, Yacine Rezgui, and Alan Kwan, “Domestic Energy Consumption Patterns in a Hot and Humid Climate: A Multiple-Case Study Analysis,” Applied Energy 114 (February 1, 2014): 353–65, https://doi.org/10.1016/j.apenergy.2013.09.061.
2. Cristian Ghiaus and Christian Inard, Energy and Environmental Issues of Smart Buildings, n.d.
3. A. Katili, R. Boukhanouf, and Robin Wilson, “Space Cooling in Buildings in Hot and Humid Climates – a Review of the Effect of Humidity on the Applicability of Existing Cooling Techniques,” 2015, https://doi.org/10.13140/RG.2.1.3011.5287.
4. Michael Donn, “Quality Assurance – Simulation and the Real World,” in Conf. Proc. of the IBPSA ’99 Buildings Simulation, Vol. III, 1139 - 146, 1999.
5. Steven J Emmerich, W Stuart Dols, and James W Axley, “Natural Ventilation Review and Plan for Design and Analysis Tools” (Gaithersburg, MD: National Institute of Standards and Technology, 2001), https://doi.org/10.6028/NIST.IR.6781.
6. Günsu Merin Abbas and Ipek Gursel Dino, “A Parametric Design Method for CFD-Supported Wind-Driven Ventilation,” vol. 609, 2019, https://doi.org/10.1088/1757-899X/609/3/032010.
7. R. Ramponi and B. Blocken, “CFD Simulation of Cross-Ventilation for a Generic Isolated Building: Impact of Computational Parameters,” Building and Environment 53 (July 1, 2012): 34–48, https://doi.org/10.1016/j.buildenv.2012.01.004.
8. Omar S. Asfour and Mohamed B. Gadi, “A Comparison between CFD and Network Models for Predicting Wind-Driven Ventilation in Buildings,” Building and Environment, Indoor Air 2005 Conference, 42, no. 12 (December 1, 2007): 4079–85, https://doi.org/10.1016/j.buildenv.2006.11.021.
9. Joseph M. Horan and Donal P. Finn, “Sensitivity of Air Change Rates in a Naturally Ventilated Atrium Space Subject to Variations in External Wind Speed and Direction,” Energy and Buildings 40, no. 8 (January 1, 2008): 1577–85, https://doi.org/10.1016/j.enbuild.2008.02.013.
10. G Carrilho da Graça et al., “Simulation of Wind-Driven Ventilative Cooling Systems for an Apartment Building in Beijing and Shanghai,” Energy and Buildings 34, no. 1 (January 1, 2002): 1–11, https://doi.org/10.1016/S0378-7788(01)00083-4.
11. Serokell, “Artificial Intelligence vs. Machine Learning vs. Deep Learning: What’s the Difference,” Medium, November 23, 2020, https://medium.com/ai-in-plain-english/artificial-intelligence-vs-machine-learning-vs-deep-learning-whats-the-difference-dccce18efe7f.
12. Sebastian Raschka and Vahid Mirjalili, Python Machine Learning (Packt Publishing Ltd, 2017).

13. Jason Brownlee, “A Gentle Introduction to Generative Adversarial Networks (GANs),” Machine Learning Mastery (blog), June 16, 2019, https://machinelearningmastery.com/what-are-generative-adversarial-networks-gans/.

14. Jonathan Hui, “GAN — What Is Generative Adversarial Networks GAN?,” Medium, December 26, 2019, https://jonathan-hui.medium.com/gan-whats-generative-adversarial-networks-and-its-application-f39ed278ef09.

15. Jason Brownlee, “How to Evaluate Generative Adversarial Networks,” Machine Learning Mastery (blog), August 25, 2019, https://machinelearningmastery.com/how-to-evaluate-generative-adversarial-networks/.

16. phpnewbie1979, “ORB Feature Point Extraction Algorithm Test,” Programmers Think, May 9, 2019, https://programmer.ink/think/orb-feature-point-extraction-algorithm-test.html.

17. Pranjal Datta, “All about Structural Similarity Index (SSIM): Theory + Code in PyTorch,” Medium, March 4, 2021, https://medium.com/srm-mic/all-about-structural-similarity-index-ssim-theory-code-in-pytorch-6551b455541e.

18. Z. Wang et al., “Image Quality Assessment: From Error Visibility to Structural Similarity,” IEEE Transactions on Image Processing 13, no. 4 (April 2004): 600–612, https://doi.org/10.1109/TIP.2003.819861.

19. Layout5 - 2020 AEC Tech Hackathon Presentation, 2020, https://www.youtube.com/watch?v=whBvHrusXBE&ab_channel=COREstudioatThorntonTomasetti.

20. Kacper Radziszewski and Marta Waczyńska, “Machine Learning Algorithm-Based Tool and Digital Framework for Substituting Daylight Simulations in Early-Stage Architectural Design Evaluation,” in Proceedings of the 2018 Symposium on Simulation for Architecture and Urban Design (SimAUD 2018) (2018 Symposium on Simulation for Architecture and Urban Design, Delft, Netherlands: Society for Modeling and Simulation International (SCS), 2018), https://doi.org/10.22360/SimAUD.2018.SimAUD.001.