Deep learning approach for predicting pedestrian dynamics for transportation hubs in early design phases

Jan Clever, Jimmy Abualdenien, André Borrmann
Chair of Computational Modeling and Simulation, Technical University of Munich, Germany
jan.clever@tum.de

Abstract. A seamless integration of model analysis and simulations into the design process is a key for supporting the different decisions, including deciding upon the position, dimensions, and materiality of building elements. Such design options are explored from the early design phases, where a decision is taken based on their performance. A crucial analysis that is necessary for the different types of buildings, especially transportation hubs, is pedestrian flow dynamics, as it evaluates the occupants' comfort and ability to evacuating the building in case of emergency. Currently, analysing pedestrians' flow is decoupled from the BIM-authoring tools, requires multiple manual steps, and is time consuming. Hence, this paper proposes a framework that leverages the latest advancements of Deep Learning (DL) for replacing pedestrian dynamics simulations by an DL model providing intermediate feedback. In more detail, a representation of the building model, including simulation parameters, is proposed as input and a Convolutional Neural Network (CNN) architecture is developed and trained to predict pedestrians' flow density heatmaps and tracing maps.

1. Introduction

The *Architecture, Engineering and Construction* (AEC) industry is a multidisciplinary sector comprising of various interconnected domain experts. During the design process of a building, each discipline makes multiple design decisions, influencing the resultant design and its performance. Over the last decade, the *Building Information Modeling* (BIM) methodology has gained popularity in fostering collaboration among the project participants and informing the design process from the early phases (Borrmann *et al.*, 2018).

Through the design phases, building models are gradually refined from a rough conceptual design (where many uncertainties are present) to highly complex individual components. In the early design phases (conceptual and preliminary phases), BIM models are subject to multiple changes in the detailed design phases (Knotten *et al.*, 2015). However, changes in the design require a relatively lower cost and efforts (Abualdenien *et al.*, 2019). Typically, architects and engineers explore and evaluate the performance of multiple design options through the comparison of their simulation results. Evaluating a design's performance involves numerous simulations and analysis. Most popularly, analysing the structural system, embodied and operational energy during the life-cycle (Abualdenien *et al.*, 2020), as well as the comfort and evacuation of occupants, a.k.a. pedestrians. Using BIM, the different objects (such as walls, stairs, and zones) can be identified, where each instance has a geometric representation and carries a set of properties (Abualdenien *et al.*, 2019). Such capabilities provide the necessary means for establishing a smooth workflow between BIM-authoring tools and simulators, where customized simulation information can be included in the model. To work independently of a particular software vendor, a variety of the existing authoring tools and simulators support the exchange of models using the open standard *Industry Foundation Classes* (IFC)[1]. Multiple researchers have investigated and proved the capabilities of using IFC BIM models as basis for simulations (Mirahadi *et al.*, 2019).

[1] https://web.archive.org/web/20111024102519/http://buildingsmart-tech.org/implementation/implementations/plominoview.allapplications/?widget=BASIC (visited: 15.03.2021)

In general, integrating simulations into an early design phases can support the decision-making process, which assists in achieving the intended project goals (Abualdenien *et al.*, 2019). Since pedestrians' behavior is essential in normal and panic situations, and highly dependent on the environment (Low, 2000), circulation routes require a special attention during the design process of a building. Therefore, this paper aims for improving the existing workflows for integrated pedestrian simulations into the design process, especially for public buildings, such as train stations.

Typically, the results of pedestrian simulations provide visibility regarding the pedestrians' comfort, circulation, and evacuation in case of emergency. However, the current state of practice involves multiple steps, including exporting building models from the BIM-authoring tool, importing them into the simulator, performing the simulation, and finally, generating a summary of the simulation results. As in addition, agent-based pedestrian simulations require high computational effort and thus long computation times, the entire process is time consuming and error-prone (Andriamamonjy *et al.*, 2018), hindering the interactive exploration of the design space. To overcome this limitation, this paper proposes a framework that leverages *Deep Learning* (DL) methods to facilitate a real-time prediction of pedestrians' comfort and circulation. More specifically, *Machine Learning* (ML) approaches can be used to avoid time-consuming simulations by supporting or even replacing them with predictive tools (Kim *et al.*, 2019). We make use of the rich information provided by BIM models as input for the ML model, thus allowing a direct interaction between creating design options and evaluating them for pedestrian dynamics performance.

This paper is organized into several parts: section 2 introduces background knowledge and related work. In section 3, the concept of our approach is described stepwise, while section 4 presents the outcome. In section 5, a conclusion sums up our results and gives an outlook to future steps.

2. Background and Related Work

2.1 Performance-based Building Design

Designing a building requires many different steps and, hence, considers multiple dependencies on decisions. Therefore, performance-based building design is a crucial method to reduce critical changes to be done in the final phase and maximize a building's performance (Mehrbod *et al.*, 2020). Furthermore, to create reliable results, sufficient data and information must be provided. Especially in early design phases, decisions can influence later performance and cost (Østergård *et al.*, 2016). To improve decisions in the design phase, BIM-based approaches were developed to use the BIM models in the process. In this manner, the authors of Röck *et al.*, 2018 integrate parts of the Life Cycle Assessment (LCA) into BIM by considering the building's materials. In this way, the designer is informed about the chosen materials' potential effects for their embodied energy. Furthermore, Hamidavi *et al.*, 2020 proposes a BIM-based optimization evaluation of a building's structural design. This approach helps to enhance the coordination between architects and structural engineers during the design phase.

2.2 Pedestrian Dynamics Analysis and Simulation Models

The functionality especially of public buildings such as train stations or shopping centres is essential in an emergency evacuation (Løvås, 1994). Moreover, pedestrian dynamics analysis is an essential aspect for efficient crowd routing concerning safety and comfort. That is strongly dependent on the shape of the building (Hanisch *et al.*, 2003). Observations show that individual pedestrians tend to choose polygon-shaped routes, following straight paths to walk on as long as possible in association with visibility. Even though some areas may be crowded, longer travelling times and unknown detours are accepted deliberately or unknowingly (Helbing *et al.*, 2001). Without being externally planned, the crowds' resulting self-organisation is somewhat based on subconscious than communication or expressed strategy, especially with unidirectional pedestrian flows (Helbing *et al.*, 2005). Besides, single persons appear to adjust their walking speed when meeting moving crowd groups within a generally crowded area. Simultaneously, individuals interpret stationary groups as clear obstacles leading to a change in their walking routes (Yi *et al.*, 2015).

Concerning simulation models, three general approaches are distinguished to model pedestrian behaviour, depending on the number of virtual pedestrians (agents). Microscopic methods define the reaction of individual agents, while macroscopic approaches model group behaviour. Furthermore, between these two approaches mesoscopic models provide information about individual agents while staying capable of handling more extensive groups (Ijaz *et al.*, 2015). Because only rule-based approaches appeared to be insufficient (Yang *et al.*, 2020), a more generalized force model was developed in Helbing *et al.*, 2000, known as the *social force* model. In principle, individual agents' repulsive interaction forces take into account other agents and obstacles while moving with a certain velocity.

In contrast to individual behaviour, crowds' demeanour is rather understood as a flow mechanism, ignoring the environment and individual interactions of agents. More specifically, the underlying idea follows the principle of continuum theory proposed by Hughes, 2002. Again, in Yang *et al.*, 2020, other approaches are introduced, like the aggregate dynamics model based on fluid dynamics. Furthermore, to simulate pedestrian crowds' multiple intentions, the potential field model works with navigation- or guidance fields. Due to strict cellular automata structuring, higher pedestrian densities or not completely cell-filling obstacles can lead to a lower representation of reality (Biedermann *et al.*, 2016). To overcome issues like these, hybrid models consider different modelling approaches for particular areas or regions evoking unique behaviour (Biedermann *et al.*, 2021). Another well-known approach is the *optimal steps model* (OSM). Instead of restricting the model to dense crowds or rigid spatial grid only, the authors of Seitz *et al.*, 2012 provide continuous space and free the agents from a strict cell-representation while keeping the stepwise movement in a discretized manner.

2.3 Train Stations and Crowd Dynamics

Concerning waiting areas in train stations, pedestrians tend to uniformly distribute over the respective spaces (Helbing *et al.*, 2001). Furthermore, observations have shown that waiting pedestrians can have a considerable influence on crowd dynamics in train stations. As a result, the walking time of arriving train passengers may increase up to 20%, leaving the platform being influenced by waiting pedestrians as well as by awkwardly positioned attraction points (Davidich *et al.*, 2013). Looking closer at different building elements, Ma *et al.*, 2013 investigated the influence of fences and pillars as separation modules in crowded areas, notably train stations. They point out the increase of pedestrians' flow rate for non-unidirectional movements when using pillars instead of other modules or none at all. Likewise, similar behaviour could be examined by Frank *et al.*, 2011, who showed an improvement in evacuation time for exit areas with pillars placed close to them.

2.4 Deep Learning Methods

In the previous paragraphs, the complexity of pedestrian behaviour and the resulting simulation models could be emphasized. Consequently, pedestrian simulations for complex building structures lead to a considerable increase in computation time. To reduce computation time, AI methods are increasingly considered by the research community. Machine learning (ML) approaches as one specific category of AI methods allow to replace time-consuming simulations by predictive methods. The concept is also known as finding and applying a surrogate function. DL methods became popular to deal with complex problems and different types of data. Various architectures of *Artificial Neural Networks* (ANNs or simply NNs) accomplished different success rates in tackling different kind of tasks, such as detection and segmentation of objects in images or natural language processing.

As a fundamental feedforward NN, commonly, the *Multilayer Perceptron* (MLP) is used for various problems. Here, single values are stored within connected computational nodes organized in (hidden) layers and processed in one direction. The choice of the number of layers is one crucial part of establishing an individual NN suitable for solving a given task. The network's principle is mapping a given input to the desired output, that is to say, a classified label. During the network training, a backpropagation algorithm optimizes the network parameters and, thus, the networks output's accuracy (Nielsen, 2015).

To better deal with images in the form of matrices, *Convolutional Neural Networks* (CNNs) achieved a remarkable success. This kind of feedforward NN consists of several layers, each performing a set of computations. First, a kernel applies a convolution operation to the input matrix that results in a so-called *feature map*. Here, the kernel can be compared to a filter, while different kernels can compute multiple feature maps in parallel within one layer yielding a feature set. Next, a nonlinear activation function like the *rectified linear unit* (ReLU) function is applied to each feature map element. In a final step, the matrix dimensions can be reduced by a pooling operation, known as down-sampling, for instance, maximum pooling. This modification lowers the computational effort of the following layer. Moreover, CNNs can pick out and also detect patterns (features) within a given dataset (Goodfellow *et al.*, 2016).

To train a neural network, a sufficient amount of data is needed. Furthermore, optimization techniques can improve the training process of the network. Providing fewer data can lead to underfitting, while overfitting may occur by using the same training data too often and, thus, the network focuses intensively on these specific examples. Overfitting is why regularization methods like the dropout can enhance the network's computations by simply varying the activated nodes almost randomly. This way, a forced uncertainty is brought into the model, and co-adaptions can be prevented and, thus, overfitting can be reduced (Srivastava *et al.*, 2014). Batch normalization was discovered being useful for strengthening a network's training process (Santurkar *et al.*, 2018). Each layer's inputs are normalized before being passed on to the corresponding activation function in the following computational nodes. Consequently, the downside known as covariate shift is decreased and deep dependencies between multiple layers are relaxed. Besides, the need of regularization methods like dropout in a network may be reduced by integrating batch normalization (Ioffe *et al.*, 2015).

CNNs are a specific ML method particularly tailored for applications in image analysis. For instance, CNNs are able to detect and distinguish cell particles from non-cell particles (Nishida *et al.*, 2018). In Brunton *et al.*, 2020, an ML approach is presented that improves optimization and performance and flow control of calculations in fluid dynamics. Another example is an ML component-based approach supporting estimating a building's heating- and cooling energy (Geyer *et al.*, 2018). Moreover, the authors state an additional benefit of improving the understanding of complex energy calculations for specific parameters.

3. Methodology

The hypothesis of this paper is that deep learning methods can understand the relationship between building information and simulation results, making it possible to replace simulations by real-time predictions. To achieve this, there are two main aspects that need to be identified: (1) how can the geometric and semantic information of the design be represented? (2) What type of simulation results are we trying to predict? The answers of these questions have a high influence on which neural network architecture is suitable, including which operations must be applied on the different layers.

As this paper aims for replacing simulation results, it proposes a framework for an automatic generation of a training dataset as well as predicting the simulation results directly from the BIM representation and simulation parameters. Being part of the workflow shown in Figure 1, a parametric model was developed that is capable of generating a variety of train station models. The train station models include additional parameters that are necessary for performing the pedestrian simulation. Then, each BIM model is exported into IFC, where the geometry and semantics are processed to generate a simulation project file. In this paper, we generate project files that are of the same structure as of the crowd simulator *Crowd:it*[2]. Crowd:it uses the optimal steps model (OSM) (Seitz *et al.*, 2012) for simulating the pedestrian's behaviour. Afterwards, since the simulation parameters are already included in the BIM model, the simulation can run automatically with no manual interaction. Once the simulation is completed, the results are post-processed to produce density heatmaps, path traces, and evacuation times. This process is automatically repeated for design variant that is generated from the parametric model. The generated dataset of BIM models and simulation results is then used to train a neural network.

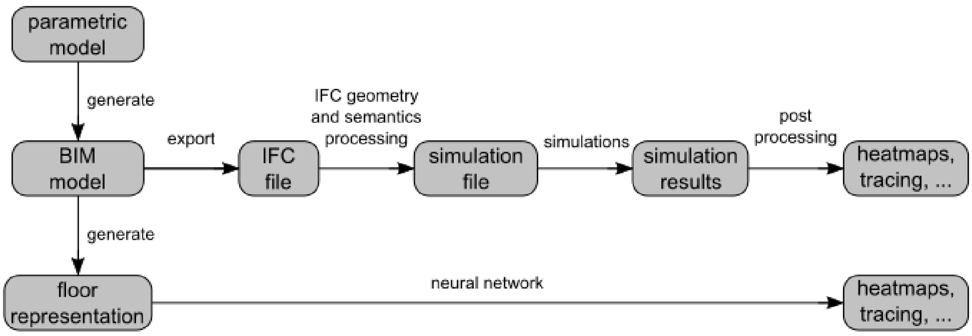

Figure 1: Workflow - conventional way vs. DL approach

3.1 Parametric Models

We developed a parametric model that allows an easy access to different model parameters for variation in the train station models, presented in Figure 2. Now, geometric parameters like the station's length, the platform's width, or the number of escalators can be easily adjusted in the BIM model without tremendous effort. In general, the number of datasets available is crucial for the training of a neural network. In our first attempt, we established in total 432 variations of generic train stations. The corresponding variation parameters are listed in Table 1.

[2] https://www.accu-rate.de/en/software-crowd-it-en/

Table 1: Parameter values for generic train station variation

Abbreviation	Meaning	Variations
F	Number of floors	2
T	Distance between train tracks	15, 25
W	Number of tracks	2, 3, 4, 5
L	Station length	150, 200, 250, 300
H	Floor height	15, 25
E	Number of escalators	1, 2, 3
P	Number of agents (per passenger coach)	5, 20, 50

In addition to the variations, specific semantic information has to be set for the different objects within each train station to ensure an automatic processing of the model by the pedestrian simulation software. In particular, special zones must be marked in the model that, for example, marking agents' spawn areas and destination. Moreover, the number of agents and a mapping of the object types to the simulation object types has to be also specified. Figure 2 shows an example of a parametric platform with four track lines, three escalators at each side, an elevator box in the middle, and two columns in between the track lines. Such building elements are translated into boundaries in the pedestrian simulator.

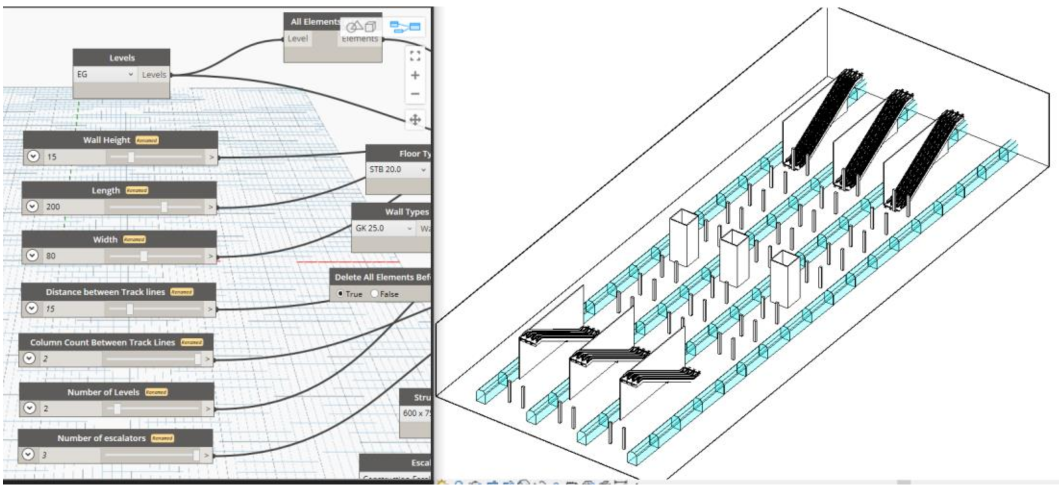

Figure 2: Tool to vary parameters (l.) that create a generic train station (r.)

The toolset used to develop this parametric model are *Autodesk Revit*[3] and *Dynamo*[4]. In this regard, the Deutsche Bahn RIL[5] guidelines were investigated and transformed into logical code that is embedded in the dynamo graph. Such parametric model provides an adaptive train station design, where changing a parameter automatically propagates to the other parameters and regenerates the station design. For the purpose of this paper, as shown in Table 1, all the models were prepared with only two floors. The scenarios we are experimenting with expect that

[3] https://www.autodesk.com/products/revit/overview?term=1-YEAR
[4] https://www.autodesk.com/products/dynamo-studio/overview
[5] https://www1.deutschebahn.com/sus-infoplattform/start/regelwerk

pedestrians will enter the train station via the train coaches and walk to the upper floor. The pedestrians choose the escalators as transition areas to reach the destination zone at the next floor directly after the end of each escalator. In each simulation run, the paths of the pedestrians are chosen according to the simulator's internal logic. Each simulation will end as soon as the last agent has reached the destination zone.

3.2 Floor Representation and Neural Network

To provide an understandable representation of the different object types for training a neural network, we propose the combination of a colour-labelled floorplan and a vector of meta-data (represented by variation parameters in Table 1). For instance, spawning zones are marked in pink while walkable areas are coloured with white, see Figure 3. As the corresponding output, the simulator crowd:it post-processes the simulation results and produces mean density heatmaps (i.e. average of agents per area) and tracing maps according to the selected routes by the agents. Figure 4 depicts an example of generated heat map is illustrated, where mean densities are coloured in blue, the darker the colour, the higher the mean density (brighter zone colour in spawn zones). The agents' traces in orange colour can be seen in Figure 4.

Figure 3: Floorplan representation with colouring

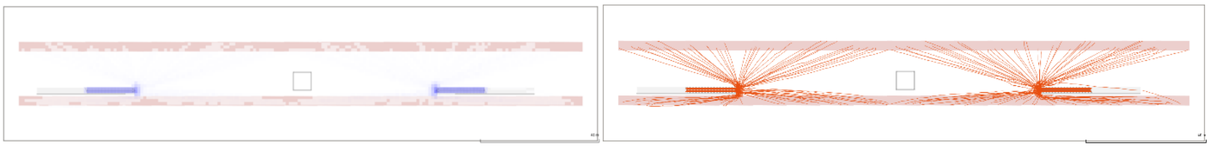

Figure 4: Heat map (l.) and tracing map (r.) examples with 5 agents per passenger coach

4. Neural Network Architecture

Although many different approaches for applying ML in this context are conceivable, in this paper, the focus is on using an image representation as an input and predict an image with densities and traces as output. Hence, we build upon the architecture of *U-Net* (Ronneberger *et al.*, 2015), a fully convolutional network, where pooling operators are replaced with upsampling operators, which improves training performance and the resolution of the output. Additionally, U-Net implements skip connections between the layers and then combines them with a concatenation layer.

Our implementation extends the U-Net architecture by an additional input layer for the meta-data that includes the station dimensions and the pedestrian simulation parameters. In this regard, the placement of the meta-data input layer should be carefully done to avoid encountering the *Vanishing Gradient* problem (Hochreiter, 1998). We optimize our network using minibatch SGD and we apply the Adam solver (Kingma *et al.*, 2014), with a learning rate of 0.002, and momentum parameters $\beta1 = 0.5$, $\beta2 = 0.999$, following the recommendations provided by Isola *et al.*, 2017. At inference time, we apply dropout and batch normalization (Ioffe *et al.*, 2015). Figure 5 presents the network architecture. It expects images with a resolution of 1024 * 1024, and produces images with the same size. In between, there is a set of downsampling and upsampling operations are extracting the different features from the image. In the middle, right after flattening the image, the second input of the meta-data is provided and concatenated with the extracted features. The lines between the sampling operations point to the concatenated features passed from each side.

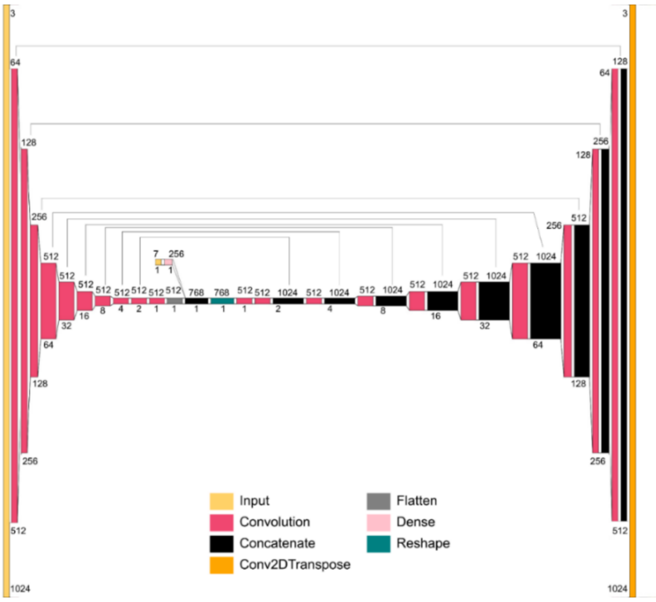

Figure 5: Neural network architecture

5. Neural Network Results & Evaluation

The training process started by splitting the dataset into training and testing. The dataset size is 432 projects with their simulation results, where 20% (87 projects) were used for testing. Before starting the training process, we have applied data augmentation, including resizing, cropping and rotating to double the amount of training data to 690 projects. To the ensure the model performance during training, 20% of the training data was used for validation in every epoch. The training used a batch size of four and ran for 300 epochs. The loss function used to quantify the quality of the predicted heatmaps and traces in comparison to the ground truth during training and validation we used the *mean absolute error* (MAE) per pixel (Asamoah *et al.*, 2018). Figure 6 shows the MAE per pixel of both, training and validation datasets for the training on generating images with heatmaps. In this regard, the error on both sets became less than 0.05 relatively fast (after few epochs). From our observations during training, we noticed that from epoch 20 the predicted images started to generate heatmaps over the right position, however, the density of those heatmaps was low. At epoch 300, the density of the generated heatmaps became fairly comparable to the ground truth by the human eye. Which highlights the need for human's perception in addition to the MAE per pixel to identify the quality of the predictions.

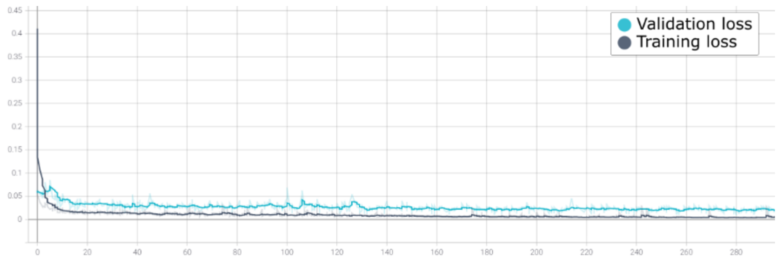

Figure 6: Heatmap – MAE per pixel

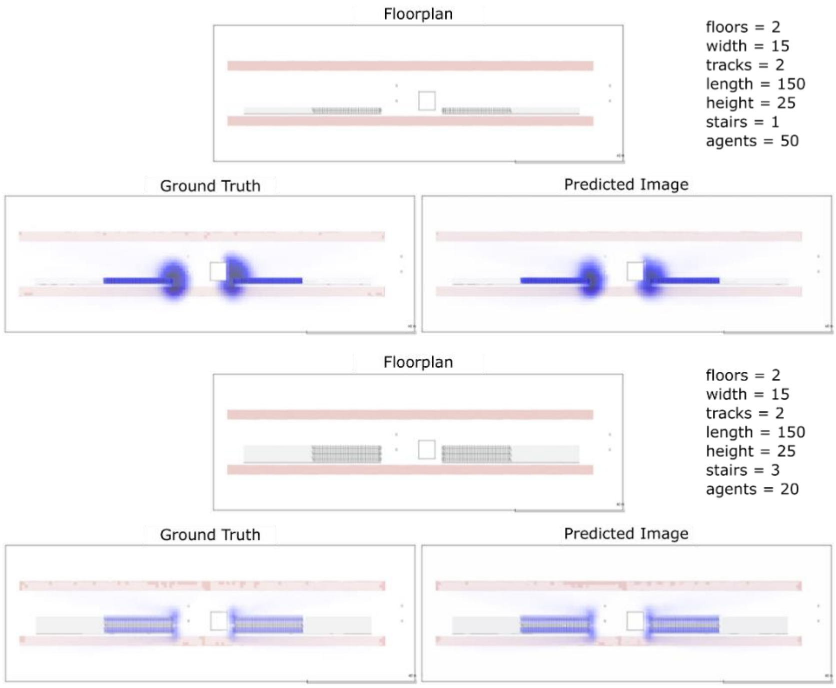

Figure 7: Heatmap – results case 1 (t.) and case 2 (b.)

The prediction of images from the test set are shown in Figure 7, comparing the input floorplan and meta-data, ground truth, and the predicted image. The predicted image in the first case has a similar overall distribution, however, the density at the start of the right stair is less than the ground truth. While in the second case, the predicted image has a slightly denser heatmap than the ground truth. Afterwards, the same process was repeated for training the network on tracing maps, with the same network parameters and loss function. As tracing maps include detailed lines for the different pedestrians, the MAE per pixel is higher than in the case of heatmaps (see Figure 8).

The predicted tracing maps from the test set are shown in Figure 9, comparing the input floorplan and meta-data, ground truth, and the predicted image. In both cases, the network was able to predict reasonable patterns that are close to the ground truth. However, similarly to the heatmaps, the densities deviate.

Overall, the network was able to understand the relation of the input (floorplan + meta-data) to the simulation results (heatmaps and tracing maps). This is shown by predicting different results for different stairs width and number of pedestrians. However, as shown in the training and validation loss figures, increasing the dataset has a high potential for improving the results. Additionally, a different loss function, other than the MAE per pixel, could provide more reasonable assessment for the quality of the predicted images.

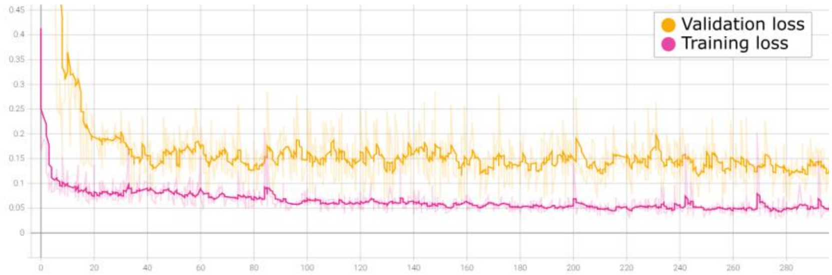

Figure 8: Tracing map – MAE per pixel

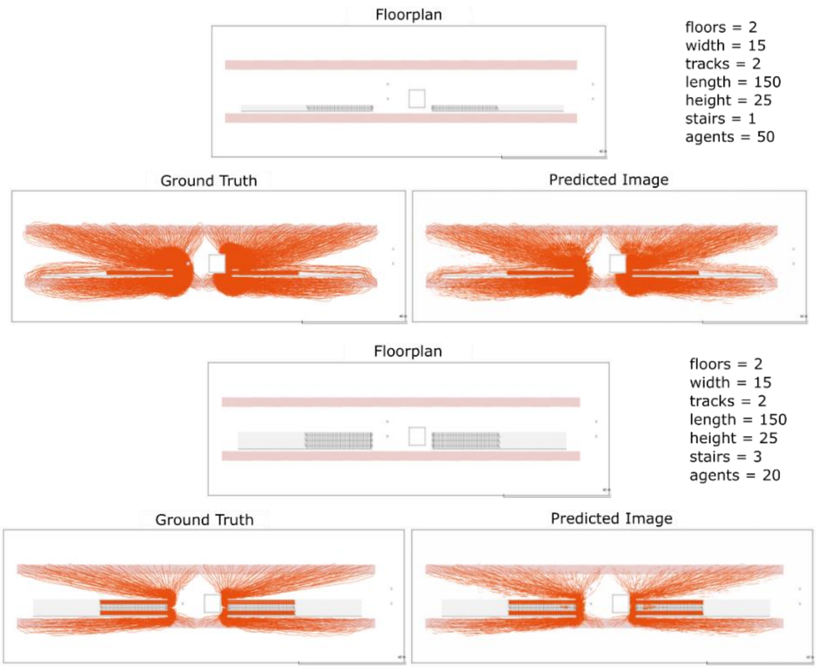

Figure 9: Tracing map – results case 1 (t.) and case 2 (b.)

6. Conclusion & Future Work

In this paper, we presented initial results of providing real-time results for a given train station geometry concerning pedestrian behaviour. Conventional pedestrian simulations can easily become very expensive in computation time. In our approach, training a CNN with image data of the BIM model, we took a first look into practical results for predicting mean densities of pedestrians and their tracing. The approach shows promising results and will be investigated further. In the first place, we clearly see the possibility of using more complex data. That is to say, generic train stations provide similar and rather simple geometric information. As a consequence, remarkable changes in the design may not be considered or understood by the network. Improvements within a predictive tool for pedestrian behavior as presented in this paper can lead to an easy access evaluation of bottlenecks caused by a building environment that is still in design. Thus, an optimal design solution can be developed with less computational effort and remarkable savings in project time.

7. Acknowledgements

We gratefully acknowledge the support of mFUND – Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) for funding the research project BEYOND.

8. References

Abualdenien, J. *et al.* (2020) 'Consistent management and evaluation of building models in the early design stages', *Journal of Information Technology in Construction*, 25, pp. 212–232. doi: 10.36680/j.itcon.2020.013.

Abualdenien, J. and Borrmann, A. (2019) 'A meta-model approach for formal specification and consistent management of multi-LOD building models', *Advanced Engineering Informatics*, 40, pp. 135–153. doi: 10.1016/j.aei.2019.04.003.

Andriamamonjy, A., Saelens, D. and Klein, R. (2018) 'An automated IFC-based workflow for building energy performance simulation with Modelica', *Automation in Construction*, 91, pp. 166–181. doi: 10.1016/j.autcon.2018.03.019.

Asamoah, D. *et al.* (2018) 'Measuring the Performance of Image Contrast Enhancement Technique', *International Journal of Computer Applications*, 181(22), pp. 6–13.

Biedermann, D. H. *et al.* (2016) 'A Hybrid and Multiscale Approach to Model and Simulate Mobility in the Context of Public Events', *Transportation Research Procedia*, 19, pp. 350–363. doi: 10.1016/j.trpro.2016.12.094.

Biedermann, D. H., Clever, J. and Borrmann, A. (2021) 'A generic and density-sensitive method for multi-scale pedestrian dynamics', *Automation in Construction*, 122, p. 103489. doi: 10.1016/j.autcon.2020.103489.

Borrmann, A. *et al.* (eds) (2018) *Building Information Modeling*. Cham: Springer International Publishing. doi: 10.1007/978-3-319-92862-3.

Brunton, S. L., Noack, B. R. and Koumoutsakos, P. (2020) 'Machine Learning for Fluid Mechanics', *Annual Review of Fluid Mechanics*, 52(1), pp. 477–508. doi: 10.1146/annurev-fluid-010719-060214.

Davidich, M. *et al.* (2013) 'Waiting zones for realistic modelling of pedestrian dynamics: A case study using two major German railway stations as examples', *Transportation Research Part C: Emerging Technologies*, 37, pp. 210–222. doi: 10.1016/j.trc.2013.02.016.

Frank, G. A. and Dorso, C. O. (2011) 'Room evacuation in the presence of an obstacle', *Physica A: Statistical Mechanics and its Applications*, 390(11), pp. 2135–2145. doi: 10.1016/j.physa.2011.01.015.

Geyer, P. and Singaravel, S. (2018) 'Component-based machine learning for performance prediction in building design', *Applied Energy*, 228(July), pp. 1439–1453. doi: 10.1016/j.apenergy.2018.07.011.

Goodfellow, I., Bengio, Y. and Courville, A. (2016) *Deep Learning*. MIT Press.

Hamidavi, T., Abrishami, S. and Hosseini, M. R. (2020) 'Towards intelligent structural design of buildings: A BIM-based solution', *Journal of Building Engineering*, 32, p. 101685. doi: 10.1016/j.jobe.2020.101685.

Hanisch, A. *et al.* (2003) 'Online simulation of pedestrian flow in public buildings', in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No.03EX693)*. IEEE, pp. 1635–1641. doi: 10.1109/WSC.2003.1261613.

Helbing, D. *et al.* (2001) 'Self-organizing pedestrian movement', *Environment and Planning B: Planning and Design*, 28(3), pp. 361–383. doi: 10.1068/b2697.

Helbing, D. *et al.* (2005) 'Self-organized pedestrian crowd dynamics: Experiments, simulations, and design solutions', *Transportation Science*, 39(1), pp. 1–24. doi: 10.1287/trsc.1040.0108.

Helbing, D., Farkas, I. and Vicsek, T. (2000) 'Simulating dynamical features of escape panic', *Nature*, 407(6803), pp. 487–490. doi: 10.1038/35035023.

Hochreiter, S. (1998) 'The vanishing gradient problem during learning recurrent neural nets and problem solutions', *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 6(02), pp. 107–116.

Hughes, R. L. (2002) 'A continuum theory for the flow of pedestrians', *Transportation Research Part B: Methodological*, 36(6), pp. 507–535. doi: 10.1016/S0191-2615(01)00015-7.

Ijaz, K., Sohail, S. and Hashish, S. (2015) 'A Survey of Latest Approaches for Crowd Simulation and Modeling using Hybrid Techniques', *17th UKSIM-AMSS International Conference on Modelling and Simulation*, pp. 111–116. doi: 10.1109/UKSim.2015.46.

Ioffe, S. and Szegedy, C. (2015) ‘Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift’. Available at: http://arxiv.org/abs/1502.03167.

Isola, P. *et al.* (2017) ‘Image-to-image translation with conditional adversarial networks’, in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134.

Kim, J. *et al.* (2019) ‘Recurrent inception convolution neural network for multi short-term load forecasting’, *Energy and Buildings*, 194(2019), pp. 328–341. doi: 10.1016/j.enbuild.2019.04.034.

Kingma, D. P. and Ba, J. (2014) ‘Adam: A method for stochastic optimization’, *arXiv preprint arXiv:1412.6980*.

Knotten, V. *et al.* (2015) ‘Design management in the building process - a review of current literature’, *Procedia Economics and Finance*, 21, pp. 120–127.

Løvås, G. G. (1994) ‘Modeling and simulation of pedestrian traffic flow’, *Transportation Research Part B: Methodological*, 28(6), pp. 429–443. doi: 10.1016/0191-2615(94)90013-2.

Low, D. J. (2000) ‘Following the crowd’, *Nature*, 407(6803), pp. 465–466. doi: 10.1038/35035192.

Ma, J. *et al.* (2013) ‘Modeling pedestrian space in complex building for efficient pedestrian traffic simulation’, *Automation in Construction*, 30, pp. 25–36. doi: 10.1016/j.autcon.2012.11.032.

Mehrbod, S., Staub-French, S. and Tory, M. (2020) ‘BIM-based building design coordination: processes, bottlenecks, and considerations’, *Canadian Journal of Civil Engineering*, 47(1), pp. 25–36. doi: 10.1139/cjce-2018-0287.

Mirahadi, F., McCabe, B. and Shahi, A. (2019) ‘IFC-centric performance-based evaluation of building evacuations using fire dynamics simulation and agent-based modeling’, *Automation in Construction*, 101, pp. 1–16. doi: 10.1016/j.autcon.2019.01.007.

Nielsen, M. A. (2015) *Neural networks and deep learning*. Determination press San Francisco, CA.

Nishida, K. and Hotta, K. (2018) ‘Robust cell particle detection to dense regions and subjective training samples based on prediction of particle center using convolutional neural network’, *PLOS ONE*. Edited by L. Wang, 13(10), p. e0203646. doi: 10.1371/journal.pone.0203646.

Østergård, T., Jensen, R. L. and Maagaard, S. E. (2016) ‘Building simulations supporting decision making in early design – A review’, *Renewable and Sustainable Energy Reviews*, 61, pp. 187–201. doi: 10.1016/j.rser.2016.03.045.

Röck, M. *et al.* (2018) ‘LCA and BIM: Visualization of environmental potentials in building construction at early design stages’, *Building and Environment*, 140, pp. 153–161. doi: 10.1016/j.buildenv.2018.05.006.

Ronneberger, O., Fischer, P. and Brox, T. (2015) ‘U-Net: Convolutional Networks for Biomedical Image Segmentation’. Available at: http://arxiv.org/abs/1505.04597.

Santurkar, S. *et al.* (2018) ‘How Does Batch Normalization Help Optimization?’ Available at: http://arxiv.org/abs/1805.11604.

Seitz, M. J. and Köster, G. (2012) ‘Natural discretization of pedestrian movement in continuous space’, *Physical Review E*, 86(4), p. 046108. doi: 10.1103/PhysRevE.86.046108.

Srivastava, N. *et al.* (2014) ‘Dropout: A Simple Way to Prevent Neural Networks from Overfitting’, *Journal of Machine Learning Research*, 15(56), pp. 1929–1958. Available at: http://jmlr.org/papers/v15/srivastava14a.html.

Yang, S. *et al.* (2020) ‘A review on crowd simulation and modeling’, *Graphical Models*, 111(October 2019), p. 101081. doi: 10.1016/j.gmod.2020.101081.

Yi, S., Li, H. and Wang, X. (2015) ‘Understanding pedestrian behaviors from stationary crowd groups’, in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, pp. 3488–3496. doi: 10.1109/CVPR.2015.7298971.