360

Through the eyes of a machine

Mapping and comparing building floor plans using unsupervised machine learning

JULIE MAKSIMOVA

UNIVERSITY COLLEGE LONDON

ABSTRACT

The latest decades have seen a growing interest in applying machine learning for solving different kinds of problems, including architectural ones. It has been used for shape generation, usage prediction, analytical tool enhancement or classification. This study is concentrated on mapping building floor plans using unsupervised machine learning. It attempts to evaluate to which extent machine learning can be used for building floor plan comparison and if the resulting distribution can be defined as an instance of machine creativity. The interest in building classifications lies in the ability of these to give light on properties that were not apparent beforehand. Consequently, a floor plan classification created by a machine could possibly look at the building from a completely different angle and unveil something new.

The research uses a dataset of computer-generated floor plans for training the machine learning algorithm. Additionally, real-life floor plans are used after the model is trained in order to evaluate its comparative abilities. The results are assessed using syntactic methods as j-graphs and mean depth. The study establishes the possibility that the algorithm is able to determine the underlying rules by which the initial dataset is created. Thus this method is useful for developing shape grammars or for generative design. This field of research is still relatively new and under-explored. This study serves as a step in understanding the place of machine learning in design process and the possibilities of its application in a creative field, such as architecture.

KEYWORDS

MAX 5 KEYWORDS (PLEASE DELETE BEFORE SUBMISSION)

Machine learning, classification, floor plans, autoencoder

1. INTRODUCTION

The interest of this research lies in the possibility of a deeper understanding of architectural form. Recent development in computational methods, specifically in architectural intelligence (AI) and machine learning (ML), has opened a window of opportunity for innovative research in all of the fields of human action, including architecture.

Buildings are primarily social structures and thus they convey meanings (Markus 1987). In order to understand or interpret these meanings classification systems are made. The criteria set for these systems depend on personal discretion of the classification creator, be it building shape, function or interior space arrangement (Ibid.). But what if the creator of that type of a system is a machine? The intention of this study is to attempt to answer the following questions:

How would a machine learning algorithm evaluate and classify a dataset of building floor plans by placing it on a two-dimensional grid and what are the potential interpretations, usages and limitations of this automatically created building floor plan classification?
To what extent could this approach be defined as an example of computer creativity and how it could be interpreted in a context of implementation of computer creativity in architectural field?

In 2001, Knight and Stiny have defined four ways in which computers support human work in the field of architecture, inventing the terms of "classical" and "non-classical" architecture, which could be deciphered respectively as "verbally explainable" and "verbally unexplainable" (Knight and Stiny 2001). In the last section, where they touch on "non-classical" computation in architecture, they admit that the work in this field remains scarce (Ibid.). ML could be considered as an example of "non-classical" computation, as the methods of its decision-making are not explicitly said or shown. It proves that work in this domain has started to attract a significant amount of researchers and architects not earlier than around 20 years ago, which shows how young this field of study is.

What really boosted the development of ML in all the fields and in architecture in particular is an exponentially increasing quantity of data (Tamke et al. 2018). The question of data is at the cornerstone of ML, as data is essential for the algorithm in order to make accurate predictions. In his book Data-Driven Design and Construction Randy Deutsch advocates for usage of big data and qualifies it as a game-changer, which would allow for the architects to create projects that respond better to human needs (Deutsch 2015, p. 29). His book and his message are anchored in the idea of beneficial potentials of data-driven design and in the belief, that computers are in fact capable of expanding our field of creative solutions. In case of a classification, the results that a machine will get would differ from a human-made classification. As any classification could give us deepened understanding of the matter that is being classified (building floor plans in this case), it is believed that a machine-made classification, being different from human-made ones, can give new insights about buildings and architecture. The novelty of this classification could even imply that the learning process of the machine and its solution finding represents an example of machine creativity (Chaillou 2019).

This paper is structured as follows: in section 2 the existing body of research in the field of the intersection of architecture and ML is going to be explored. In section 3 the detailed methodology is going to be presented. Afterwards, in section 4, the research results are going to be discussed. Finally, in section 5 conclusions are going to be drawn with the subsequent suggestion of possible continuation of the present research subject.

2. THEORY

Computers are used in architecture on various stages and for attaining various goals, from visualisation of ideas to a numerical analysis of spatial properties of a project (Knight and Stiny 2001). ML, contrary to other computational methods, grants more autonomy to the machine, as it lets the algorithms rely on its own experience and to “learn” form it. In 2018 Tamke et al. identified three established applications of ML in architecture: generative design (which concerns itself with new design idea production), shape recognition (applied for architectural space usage prediction) and categorisation (which comprises shape classification and design space exploration) (Tamke et al. 2018). Let us now explore each of these three above identified applications in details.

Generative algorithms work by attempting to find an architectural solution that approaches the goal set by the designer (Gero 1996). In 2014 the generative algorithms have seen their major development due to the advent of Generative Adversarial Networks (GANs) (Chaillou 2019). Generative design is not the main focus of this paper, however several pieces of research in this field can be noted in order to illustrate the variety of ML application of architecture and the progress of these new methods. Theses are, among others, the papers by Wu et al., Goodman and Chaillou, all published in 2019 and consisting in building floor plan generation (Wu et al. 2019, Goodman 2019, Chaillou 2019). These studies are all extremely recent which illustrates that this field is still young and experimental. On the other hand, the amount of research done in this area shows a growing interest in it.

The next example of ML application in architecture is shape recognition, which could be directly associated with usage prediction. One example of research in this area of study includes the dissertation by Boyana Buyuklieva, where she looked at the correlation between space shape and distribution and its function in offices. After testing three supervised and two unsupervised learning techniques, she could obtain 67.9% accuracy in usage prediction (Buyuklieva 2015).

Furthermore, similar approaches have been also used in the context of urban space. The research by Thirapongphaiboon and Hanna attempts to predict building land use of the corresponding street segment. Using supervised neural networks they have attained prediction accuracy of 85% (Thirapongphaiboon and Hanna 2019). As it can be noticed, both of the studies have not attained a 100% accuracy, which could be explained by the novelty of the approach.

Classification and typologies is another big theme for ML in architecture. It goes without saying, that building classifications were attracting researchers long before ML came to be. For instance, Shpuza and Peponis used this tool in order to find correlations between building shape and integration in office floor plans (Shpuza and Peponis 2008) and Sailer mapped school buildings according to the characteristics of people movement in them (Sailer 2018). Steadman went even further by developing an “architectural morphospace”, where he maps the universe of all the possible organisations of a rectangular plan (Steadman 2010). ML has a potential to enhance building classification, as it is presumably free of human bias, be it social, political or even personal, that influences conception of a classification of any kind (Markus 1987).

In 2004 Benoudjit et al. were interested in finding correlations between built form and human perception of space. They have generated a set of three-dimensional shapes, classified them using a ML algorithm and discovered several groupings (Benoudjit et al. 2004). Ten years later, Derix and Jagannath conducted a similar experiment (Derix and Jagannath 2014). Instead of three-dimensional abstract shapes they used justified graphs (j-graphs) of existing buildings. J-graphs correspond to a graph representation of space, where, in case of interior spaces, a node corresponds to a room or a convex space (a space where the whole space is seen from every point in this space) and a connection corresponds to a passage between rooms (Hillier and Hanson 1984, p. 14). Finally, in 2016 Harding combined ML with parametric design in order to visualise the universe of all possible shapes of any given parameters (Harding 2016). The three previous studies make use of the same ML algorithm, entitled Self-Organising Maps (SOM). It allows to map high dimensional data on a grid of a pre-defined size according to similarities in the data discovered by the algorithm (Benoudjit et al. 2004).

More recently, a research by Varoudis and Penn has combined shape classification and generative design using another ML technique. The aim of that study was to analyse examples of street arrangements in London and to generate a new arrangement in the style of that city (Varoudis and Penn 2019). The algorithm used an unsupervised learning technique named Variational autoencoder (VAE). It works by finding the most important features that describe the dataset. Most importantly, it generates a two-dimensional space, where the examples of the dataset are placed in relation to each other, and the blank space represents the potential form that lies in between, which allows for generation of new shape (Ibid.).

It has been seen that the application of ML techniques within architecture has been extremely varied. What characterises all of the above described studies, is that they are experimental and innovative, and their practical application has not yet been established. Often the results of these pieces of research, although promising, still require further testing in order to perfect and polish the newly found methodology before it can operate with consistency. The potential for exploration is enormous, as virtually every piece of knowledge in this field that has been obtained so far has a need for further development and deepening.

For the present study, the subject of shape classification with ML has been chosen as the key interest. It has been noticed that space categorisation by applying ML has been used to: compare and describe three-dimensional abstract shapes (Benoudjit et al. 2004), compare and describe individual spaces inside a building (Derix et al. 2014) or to analyse street arrangement for the purpose of generative design (Varoudis and Penn 2019). What has not been yet seen, is how real floor plans, or at least realistic floor plans, could be compared and classified using these techniques, which is what this study is going to attempt to solve. In addition, analytical design methods, which form a part of Space Syntax methodology, are going to be used in order to evaluate subsequent results.

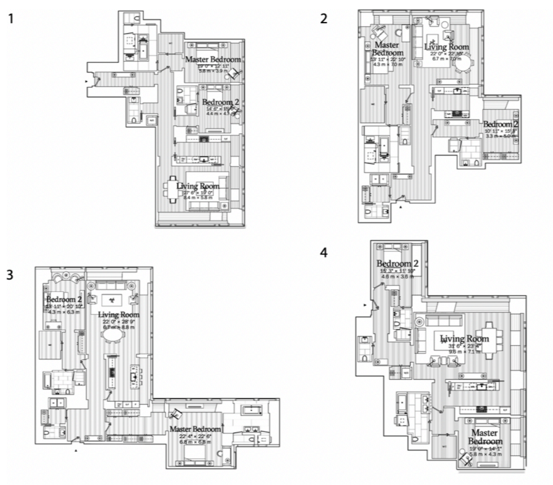
3. DATASETS AND METHODS

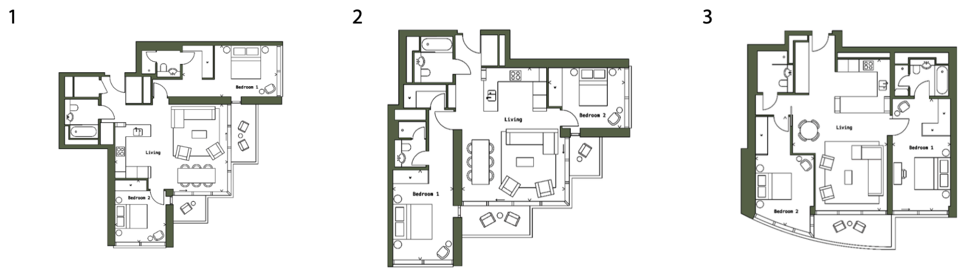
3.1. Dataset generation

In a ML problem gathering a high-quality dataset that is large enough becomes a crucial and at the same time one of the most effort-consuming and challenging parts of the process, as ML is essentially a statistical technique (Chaillou 2019). Some areas of study already possess a substantial amount of data suitable for ML, as for instance machine translation. Most of the research fields however lack data and/or the data that is available is unusable (Roh et al. 2018). In the case of this study, the dataset had to be a collection of 2D architectural plans. It has been discovered that these could be found online in image or .dwg format. However these plans are extremely heterogenous and demand an enormously time-consuming task of image pre- processing in order for the dataset to become usable for a ML algorithm (Tarabishy et al. 2020).

Consequently, if a dataset could not be procured it has to be generated (Roh et al. 2018). For that purpose, it was decided to use Grasshopper. In order to limit the scope of this study, one building type had to be selected. It was decided to concentrate on housing as it is a universal building type, which is prevalent and which every person experiences (Hanson 1998, p. 2.). Additionally, a housing building plan could be of relatively small size, avoiding to be required to generate complex shapes.

4 apartments from 53W53 by Jean Nouvel:





3 apartments from One Park Drive by Herzog & de Meuron:

2 apartments from the Barbican by Powell and Chamberlin, Bon:

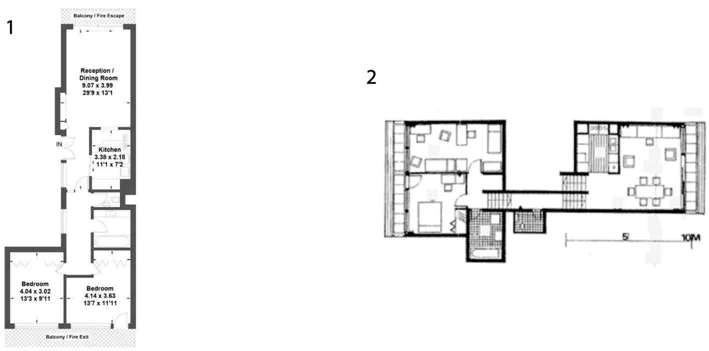


Figure 1: floor plans of 9 real-life apartments selected for the validation of the machine-made building floor plan classification.

For the dataset generation the study applies the Magnetizing plugin for Grasshopper. This plugin allows to automatically generate a floor plan, using following categories as input: room number and area of each room, which room is adjacent to which, entrance location as a point and floor plan boundary. The particularity of the plugin is that the floor plans it creates always possesses a corridor to which every room is connected, limiting the initial dataset to this layout type. The output is a collection of corridor and room boundaries. It has to be noted that the plugin produces exclusively the shape of the plan, but not its other features, as for instance entrance or doors between rooms, which have to be added subsequently.

It could be observed that the floor plans outputted by the Magnetizing plugin can be characterised by its jagged outline. As the process of this research would involve comparing computer-generated floor plans with real-life floor plans it was decided to find examples with similar characteristics, so that the ML algorithm concentrates more on finding some possible structural differences and is not distracted by a staggering contrast in shape of the generated and real-life floor plans. The choice was made to concentrate on the floor plans of apartments in apartment blocks, as these often could represent irregular shapes that are combined together as a puzzle to form a largely rectangular entity. The following examples have been selected: 4 apartments from 53W53 by Jean Nouvel, 3 apartments from One Park Drive by Herzog & de Meuron and 2 apartments from the Barbican by Chamberlin, Powell and Bon. All of these are two-bedroom apartments (fig. 1).

Apartments by Jean Nouvel were set as an example for deciding on the room number. The floor plan generating plugin was equally set to produce floor plans with two bedrooms, two bathrooms, a lavatory, a living room and a kitchen. In particular, the room areas and adjacencies were reproduced from the apartment 1 from the Jean Nouvel set: in this apartment the kitchen is directly connected to the living room and each bathroom is directly connected to the corresponding bedroom.

When the floor plan consisting of walls only is generated, the next step is adding doors between rooms as well as the entrance. For the sake of simplicity, a connection between two rooms is being represented as an opening in their common wall. The corridor and the rooms are always connected, thus the openings between them are created. Additionally, following the example floor plan that has been set previously (apartment 1 from the Jean Nouvel set), if in the resulting floor plan the kitchen and the living room and/or the bathroom 1 and the bedroom 1 and/or the bathroom 2 and the bedroom 2 have a common wall, an opening between them is automatically produced. An entrance that is also represented as an opening in the wall, is randomly placed in a corridor wall that is not adjacent to any rooms. Additionally, in order to avoid an excessively and unrealistically jagged floor plan outline, the perimeter was limited to 80 m (the perimeter of the example apartment by Jean Nouvel has a perimeter of roughly 74 m). Following these settings a dataset of 1000 floor plans has been generated (fig. 2).

Figure 2: Examples of the final floor plan dataset, which consists of a 1000 binary images of two-bedroom apartments that possess passages between rooms and an entrance.

3.2. Development of the machine learning algorithm

The core section of this study is writing and training the ML algorithm. First, the appropriate architecture had to be decided on. In section 2 it has been seen that several researchers have implemented ML in their studies. In particular, Benoudjit et al. 2004, Derix and Jagannath 2014 and Harding 2016 have used SOM and Varoudis and Penn 2019 have opted for VAE. Both SOM and VAE are useful for representing the dataset in a 2-dimensional space, although they do it differently. In a SOM this space is a grid of pre-determined dimensions, where the examples are placed on the corresponding nodes, so it is essentially a classification algorithm (Benoudjit et al. 2004). VAE on the other hand does not have these limitations, as it places the examples in the space closer or further from each other based on their similarity (Varoudis and Penn 2019). It was decided to pursue the work in VAE, so let us begin by explaining how this algorithm operates.

First let us start with a definition of an autoencoder and then let us see how a variational autoencoder differs from it. The main principle of every autoencoder is data compression and subsequent reconstruction. It consists of two parts: encoder and decoder, where the former reduces the dimensions of the data, and the latter attempts to reconstruct the initial data based on the data with reduced dimensions (Rocca 2019). Overall, it represents a neural network which in its first part (encoder) has the number of nodes diminishing on every layer, the part with the smallest number of nodes is called the bottleneck, and in the second half the node number is gradually expanding (decoder), mirroring the encoder, until it arrives to the initial node number (fig. 3).

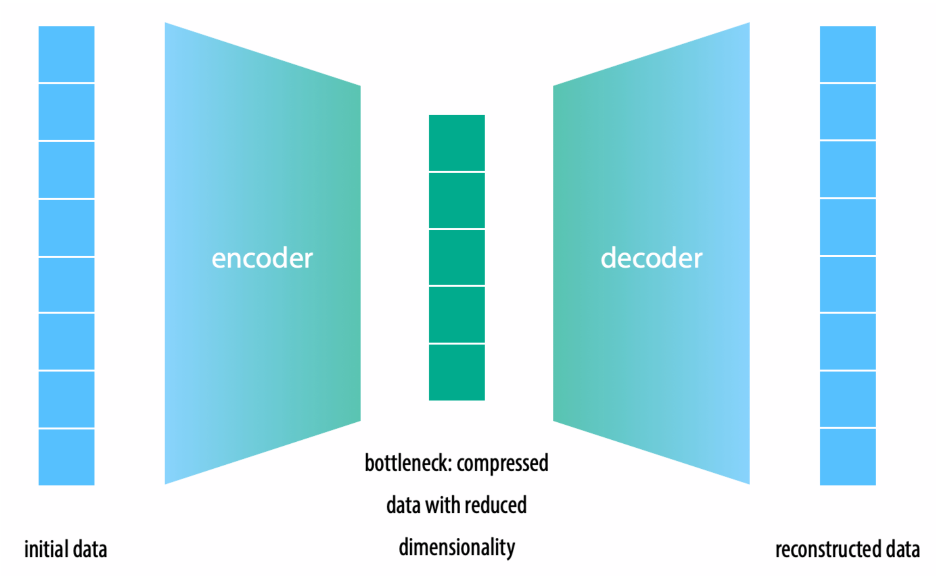


Figure 3: Autoencoder workflow: initial data is compressed by the encoder and then reconstructed by the decoder. The network is trained to reconstruct the data with the most precision possible.

The bottleneck of the autoencoder is also called the latent space, where the examples are placed according to their similarities to each other. A two-dimensional representation of this space is what is going to be used in this study. The difference between a standard autoencoder and a VAE is that in the case of VAE the latent space is more regularised so the scattering of examples is smoother, as every example is encoded not as a single point, but as a distribution (fig. 4) (Ibid.). Another result, is that the space between the points can be exploited in order to generate new shapes. Thus, VAEs are becoming more and more present in generative designs (Doersch 2016).

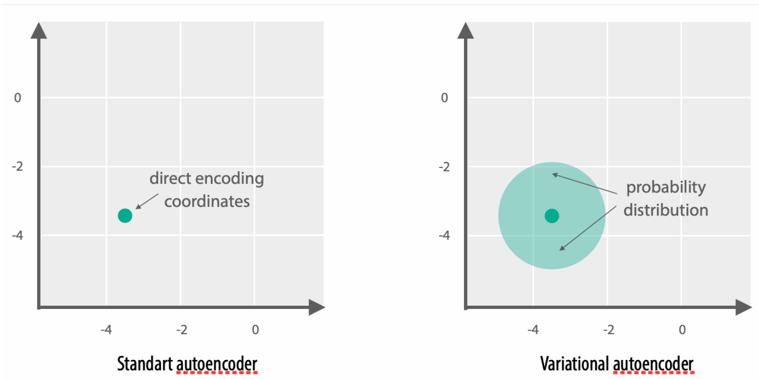


Figure 4: Difference between standard and VAE: autoencoder determines only the coordinates of an example, VAE adds to it a probability distribution, which allows to place other examples according to similarities.

Finally, the architecture of both encoder and decoder had to be decided on. For this study convolutional neural networks (CNN) are used. This is currently the most common architecture for image recognition and analysis in ML (Saha 2018). In a standard neural network every pixel value of an image is stored in a one-dimensional vector, which means that the spatial information is lost

(Ibid.). In a CNN however each subset of an image is looked at through a filter of given dimensions, where each filter learns to recognise particular features (for instance lines, shapes etc.) (Varoudis and Penn 2019). In that way spatial relations between the pixels is conserved.

The process of VAE implementation is as follows: the first step is loading the 1000 floor plan dataset generated earlier in Grasshopper. Next is the training step: the data is passed through the VAE. The algorithm is set to run for 60 epochs, as it was experimentally determined to be an optimum amount of iterations that would minimise loss as possible and avoid overfitting (fig. 5). Lastly, using the trained model the floor plan images are visualised on a two-dimensional grid. After that is done, the final stage of the research could be started.[1]

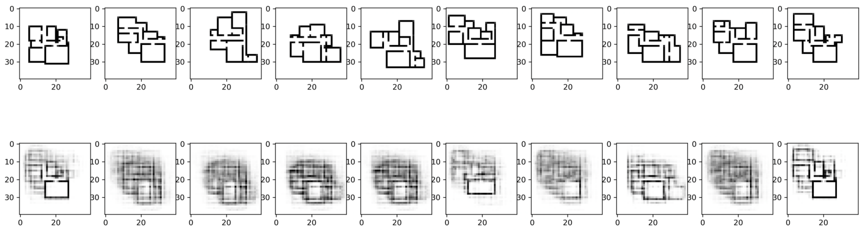


Figure 5: examples of the dataset before (top) and after (bottom) being passed through the VAE. The decoded images (on the bottom) have become blurry and quite different from the initial images.

3.3. Syntactic analysis

The analysis stage is the moment when the previous processes could finally be made sense of. This task is essential in order to understand and interpret the findings. Space Syntax is a collection of quantitative methods for describing space (Hillier and Hanson 1984, p. 66), and some of the Space Syntax techniques, described later in this section, are going to be used in order to assess the results of the ML algorithm training. During this study the syntactic analysis will be broken down in 2 stages: analysis of the two-dimensional grid, outputted by the VAE, with the generated floor plan dataset distributed on it, and the analysis of the existing building floor plans, as well as the study of the location of these floor plans on the above-mentioned two-dimensional grid. Let us explore each stage in order.

“A building may [...] be defined abstractly as a certain ordering of categories” (Hanson 1998, p. 6), which is why the whole process of syntactic analysis is going to be largely based on the justified graphs (j-graphs) method. A j-graph is a relational representation of a space (Ostwald 2011). In the case of the interior spaces, every separate room can be conceptualised as a graph node, while the passages, or doors, represent edges or connections. As it has been shown in Hanson’s study, in 2 equivalent houses, where the connections between rooms in the first one differed from the connections in the second, social differences of space usage were prominent. This difference could only be observed by studying spatial distribution, rather than concentrating exclusively on the

[1] Full code available at: https://github.com/juliemaksimova/vae_and_visualisation/tree/main

architectural form (Hanson 1998, p. 109). Therefore a j-graph representation of space becomes an extremely efficient method for understanding space. It becomes even more so when one intends to compare two or more spaces.

The j-graphs are going to be described among other in terms of their "ringiness" and "depth" (Hillier and Hanson 1984, p. 102), which are some of the most fundamental methods of assessing spatial configurations (Hanson 1998, p. 27). Ringiness can describe a permeability of a room, and in consequence indicate the control measure, which represents how many are exclusively accessible from any given space (Ibid.). Depth represents a topological distance between nodes, quantified in steps (in a case of two connected spaces, topological distance between them is equal to one) (Hillier and Hanson 1984, p. 108).

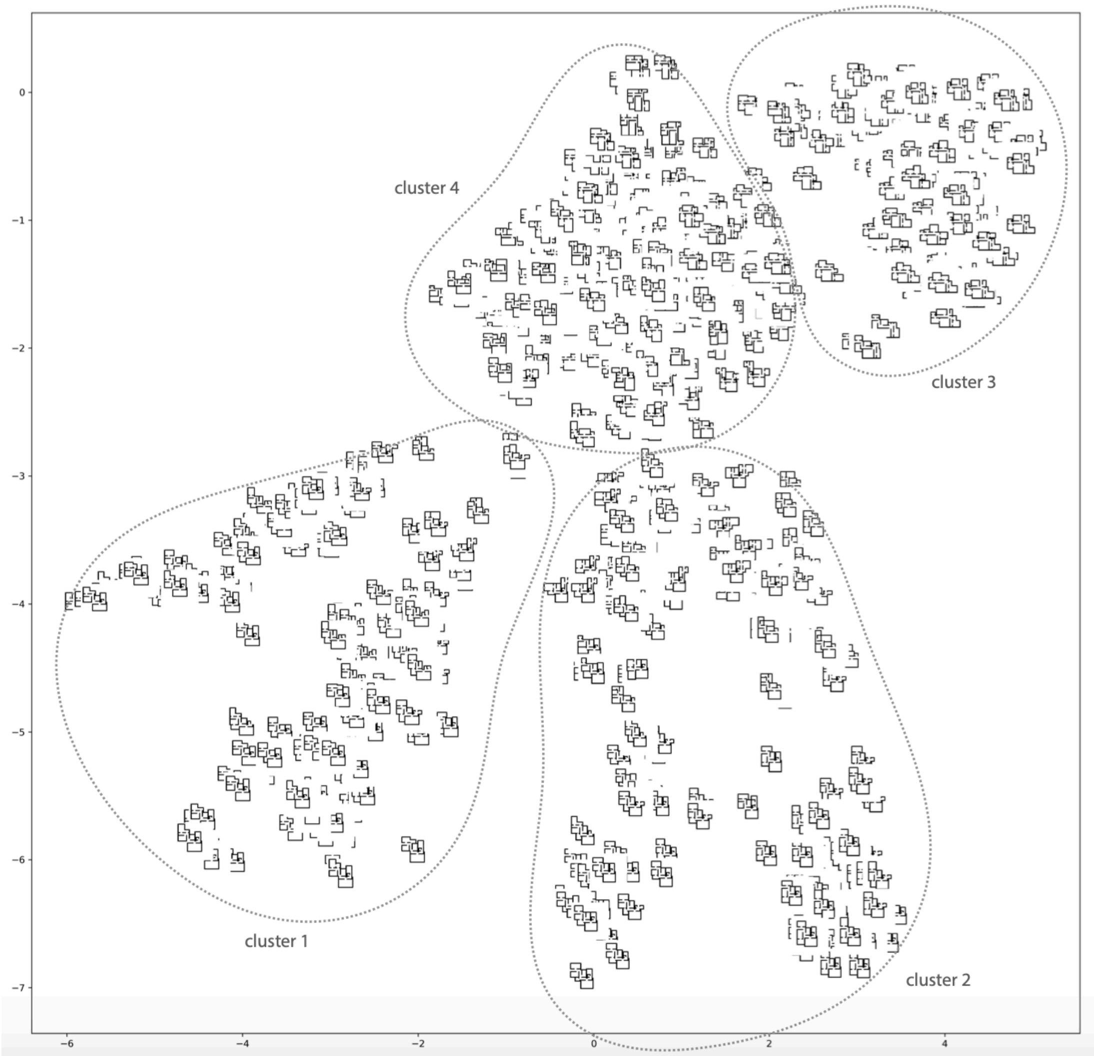


Figure 6: two-dimensional representation of the VAE latent space with the dataset representations projected on it and with the 4 clusters marked.

Mean depth of a node can be calculated by dividing the sum of all depth values from that node divided by the total number of nodes in the j-graphs minus one (Ostwald 2011). Subsequently, the mean depth values of each room in a plan can be colour coded. This is going to be useful for looking at a large amount of data at the same time in order to notice patterns. In this study, the colour coding of the floor plans in the dataset is done during the dataset generation in Grasshopper. When a floor plan is generated, its j-graph and the consequent measures, including mean depth, are calculated, and the colour-coded floor plans are saved for future usage.

4. RESULTS

Let us begin by looking at the latent space in a two-dimensional projection with the floor plans placed there. It could be visually observed that some of the images are outliers, while others are closely grouped together to form smaller clusters. The dataset distribution could be conveniently divided into 4 clusters (fig. 6).

Let us now analyse some of the clusters in details. In the clusters 1, 2 and 3 there is a noticeable resemblance between the overall shapes of the floor plans. It has to be noted that these observations are purely visual at the moment and the number of identified shapes is not exhaustive. This analysis serves as a first stage of breakdown of the result of a machine-made classification and as an attempt to intellectually comprehend it as a human being. Overall, the algorithm seems to have identified similarities between the floor plans inside these three clusters and have placed them accordingly. The cluster 4 however is located on the intersection of the 3 others and it becomes more challenging to visually determine its intrinsic logic, as it seems less homogenous as the others.

To summarise, there seems to be an observable logic of floor plan distribution, even though it is not always obvious. At the same time it has to be noted that understanding this logic can be insightful but it is not essential, as the main goal of this study is to evaluate the usefulness and inventiveness the machine-made distribution.

In order to compare how VAE classifies spaces using syntactic methods of space differentiation, all the plans had the mean depth of each room on every plan is calculated and colour coded. Mean depth number indicates how far, on average, a space in the network is from any other space. The bigger the number, the deeper the space. Depth value indicates such basic properties of space as accessibility, and thus can describe efficiently a large aspect of how a space is structured (Hillier and Hanson 1984, p. 108).

In the case of this study due to the settings of the generative algorithm the variations in floor plans structure is limited. Firstly, because the room number is limited to seven plus the corridor. Secondly, because the algorithm attempts to connect all the rooms directly to the corridor. Thirdly, because connections between rooms are also constrained. As a result, five types of rooms can be defined: where no rooms are connected to each other, one pair of rooms is connected, two pair of rooms, three pair of rooms and lastly where one of the rooms is not connected to the corridor.

Thus, the entire dataset represents different instances of these five room arrangement types. Next, the j-graphs of each room arrangement type can be constructed (fig. 7). From there the mean depth from each room in each type can be calculated as well as the average of these values which would characterise the plan as a whole.

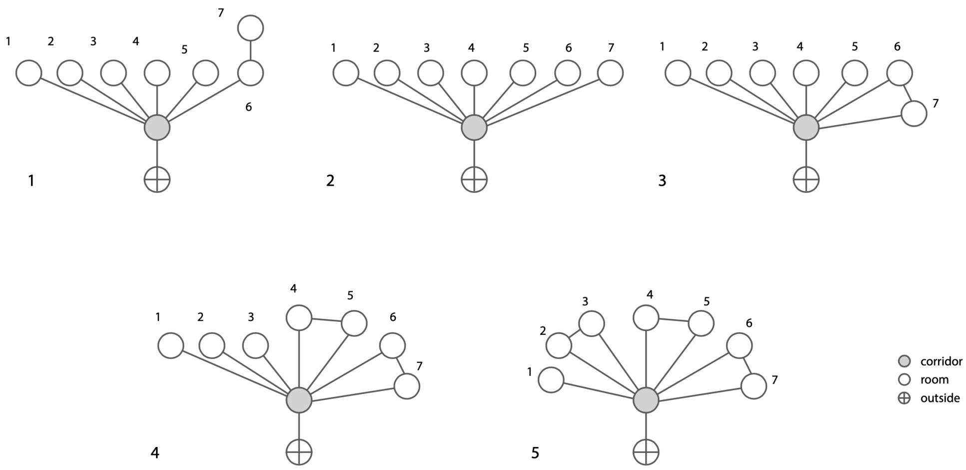


Figure 7: from left to right: 1. one room not connected to the corridor, 2. rooms not interconnected, 3. one pair of rooms connected, 4. two pairs of rooms connected, 5. three pairs of rooms connected. Room numbering corresponds to the numbers in the Table 1.

	Corridor	Room 1	Room 2	Room 3	Room 4	Room 5	Room 6	Room 7	Average
Type 1	1.142857	2	2	2	2	2	1.714286	2.571429	1.928571
Type 2	1	1.85	1.85	1.85	1.85	1.85	1.85	1.85	1.75
Type 3	1	1.857143	1.857143	1.857143	1.857143	1.857143	1.714286	1.714286	1.71
Type 4	1	1.857143	1.857143	1.857143	1.714286	1.714286	1.714286	1.714286	1.678571
Type 5	1	1.857143	1.714286	1.714286	1.714286	1.714286	1.714286	1.714286	1.642857

Table 1: mean depth values for each of the five room arrangement type. Includes the mean depth of every room as well as the average mean depth of all the rooms in every type.

It can be seen that the type 1 arrangement is the deepest one, as all its mean depth values are the highest. It becomes gradually more shallow from the type 2 to 4, with the type 5 being the shallowest, as it possesses the most rings. Now let us plot these types according to the classification that has been already cannulated by the ML algorithm.

Table 2 shows the exact number of each plan type in each cluster. Several things can be observed from there. Firstly, the cluster 4 is the largest one, having 407 examples, while the other clusters have around 200 each. Secondly, there is a differentiation in the distribution of plan types: the deepest (type 1) is found almost exclusively in the cluster 1, and the shallowest (type 5) – in the cluster 3.

	Type 1	Type 2	Type 3	Type 4	Type 5	Total
Cluster 1	15	24	147	14	0	200
Cluster 2	1	86	98	17	1	203
Cluster 3	0	6	66	96	22	190
Cluster 4	0	123	184	98	2	407

Table 2: floor plan number in each cluster according to each type

In the first two clusters there is a minimal presence of the type 4 plans and an almost complete absence of type 5 plans, which are concentrated in the cluster 3. Cluster 4 has a relatively equal share of type 2, 3 and 4 plans, which makes it difficult to see a pattern in it. However, there seems to be a pattern in the remaining clusters. The shallowest plans (type 4 and 5) are more prevalent in the cluster 3 and the deepest plans (type 1 and 2) – in the clusters 1 and 2, making it look like the mean depth of the plans gradually changes from the deepest (bottom left of the grid) to the shallowest (upper right), being in between in the centre (cluster 4).

Further ANOVA analysis (analysis of variance) shows that there seems to be indeed a correlation, returning a p value of 0.00446 which means that there is a statistically a 0.446% probability that the results are random (Table 3).

Data summary				
Groups	N	Mean	Std. Dev.	Std. Error
Group 1	4	4	7.3485	3.6742
Group 2	4	59.75	54.3346	27.1673
Group 3	4	123.75	52.1816	26.0908
Group 4	4	56.25	47.0771	23.5385
Group 5	4	6.25	10.5317	5.2658

ANOVA Summary					
Source	Degree of Freedom DF	Sum of Squares SS	Mean Square MS	F-Stat	P-Value
Between Groups	4	38413	9603.25	5.96006	0.00446

ANOVA Summary					
Source	Degree of Freedom DF	Sum of Squares SS	Mean Square MS	F-Stat	P-Value
Within Groups	15	24169.0159	1611.2677		
Total:	19	62582.0159			

Table 3: ANOVA analysis of the correlation between floor plan types and their cluster placements

Finally, let us plot the images of the floor plans with their rooms coloured according to their mean depth values (fig. 8). With the colours it is easier to see the patterns in the corridor shapes, which are characteristic to each cluster.

There is an observable prevalence of Z-shaped corridors in the clusters 1 and 2, T-shaped corridors in the cluster 3 and mixed tendencies in the cluster 4. These correlations correspond to the earlier noticed associations between the floor plan shapes and their location on the grid and between the floor plan average mean depth values and location. The meaning of this is going to be discussed in the final section, but it could be already noted that the algorithm apparently has managed to classify the dataset of floor plans according to the differences between them.

Before adding the real-life building floor plans to the distribution, it is going to be useful to syntactically analyse and compare them by measuring their mean depth, starting with the four apartments by Jean Nouvel (Table 4). Three of these apartments are structurally identical. Additionally, as these were taken as an example for the generated dataset, their room number is the same. As it can be seen from the average mean depth value, these apartments are deeper than the types 3, 4 and 5 in the generated dataset. Later in this section their placement on the grid is going to be verified in order to see, if the algorithm places them together with the more deeper examples. But first let us introduce the mean depth values of the other two floor plan sets that are used for this study: Herzog & de Meuron (Table 5) and Barbican (Table 6). These floor plans are more divergent from the initial dataset, as their room number is different.

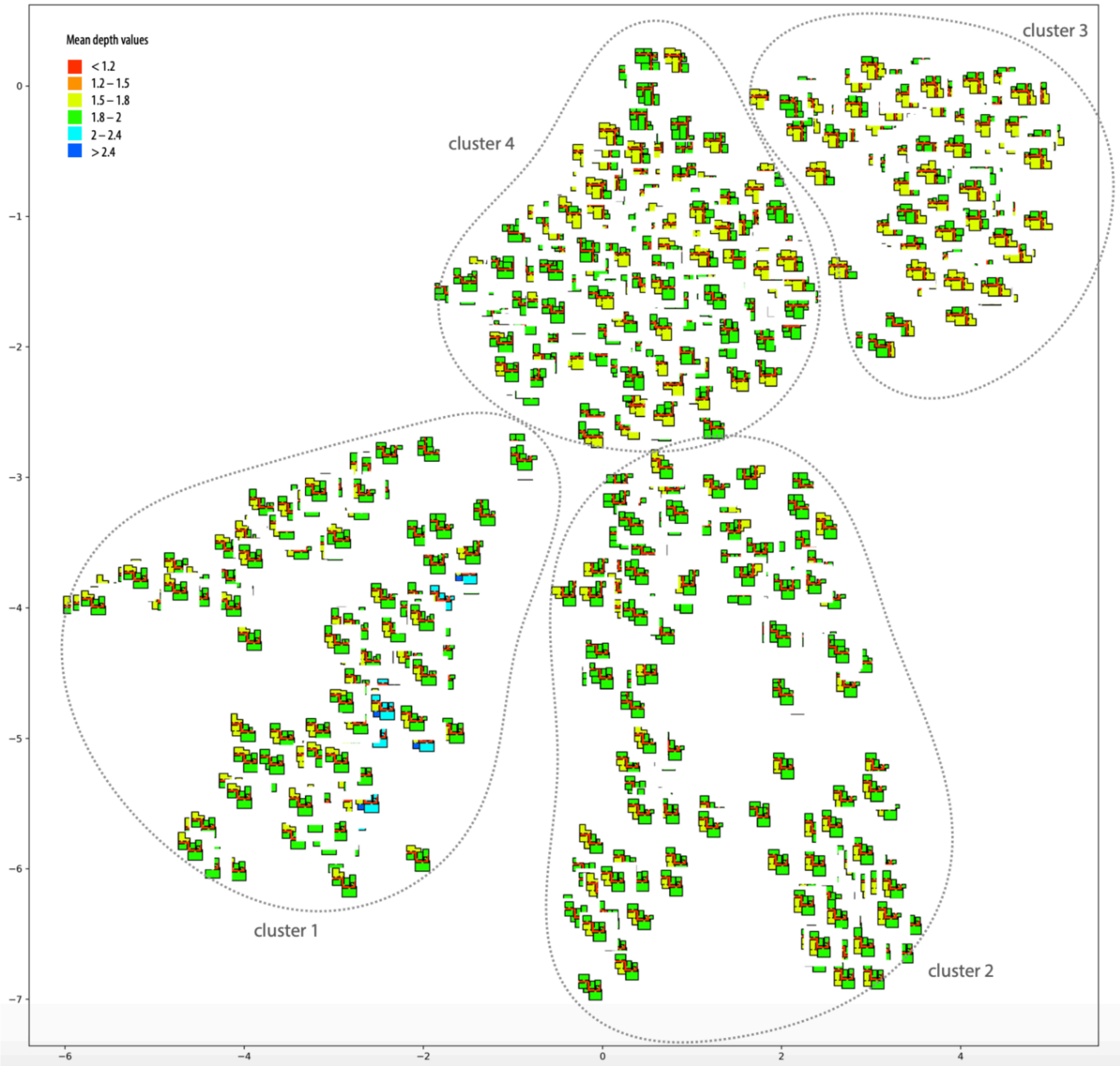


Figure 8: two-dimensional representation of the VAE latent space with colour-coded mean depth of the dataset

	Corridor	Room 1	Room 2	Room 3	Room 4	Room 5	Room 6	Room 7	Average
App. 1	1.142857	1.857143	1.857143	2	2	2	1.714286	2.571429	1.892857
App. 2	1	1.85	1.85	1.85	1.85	1.85	1.85	1.85	1.75

	Corridor	Room 1	Room 2	Room 3	Room 4	Room 5	Room 6	Room 7	Average
App. 3	1.142857	1.857143	1.857143	2	1.714286	2.571429	2	2	1.892857
App. 4	1.142857	1.857143	1.857143	2	2	2	1.714286	2.571429	1.892857

Table 4: Mean depth values of each room in every Jean Nouvel apartment

	Corridor	Room 1	Room 2	Room 3	Room 4	Room 5	Average
App. 1	0.857143	1.142857	1.428571	1.714286	1.428571	1.428571	1.333333
App. 2	0.857143	1.142857	1.428571	1.714286	1.428571	1.428571	1.333333
App. 3	1.142857	1.142857	1.428571	1.714286	2	1.714286	1.52381

Table 5: Mean depth values of each room in every Herzog & de Meuron apartment

	Corridor	Room 1	Room 2	Room 3	Room 4	Room 5	Room 6	Average
App. 1	0.857143	1.571429	1.571429	1.571429	1.571429	1.571429	1.571429	1.469388
App. 2	0.857143	1.571429	1.571429	1.571429	1.571429	1.571429	1.571429	1.469388

Table 6: Mean depth values of each room in every Barbican apartment

The next step is to project these nine floor plans onto the distribution that was generated by the ML algorithm in order to evaluate their location on it.

After adding these floor plans to the initial dataset, two things could be observed (fig. 9). First of all, all of the existing plans are situated in the cluster 4, the most heterogeneous of all, as it was determined before. Secondly, it seems that the plans from the same development are not grouped together. In fact some plans from different developments are placed closer to each other they are to the plans of the same development.

Looking purely at the shape similarities or when assessing the mean depth values in all of the cases it is difficult to find correlation between the location of the generated floor plans and the existing floor plans (fig. 10). If some similarities occur between the real-life floor plans and the neighbouring generated floor plans, these similarities are not systematic and are results of purely personal observations, not confirmed by any objective data. It is then difficult to suggest that the algorithm has placed the existing floor plans according to any logic at all. At least, this logic is not comprehendible at this level, which could also be the case, as humans and machine has a completely different reasoning.

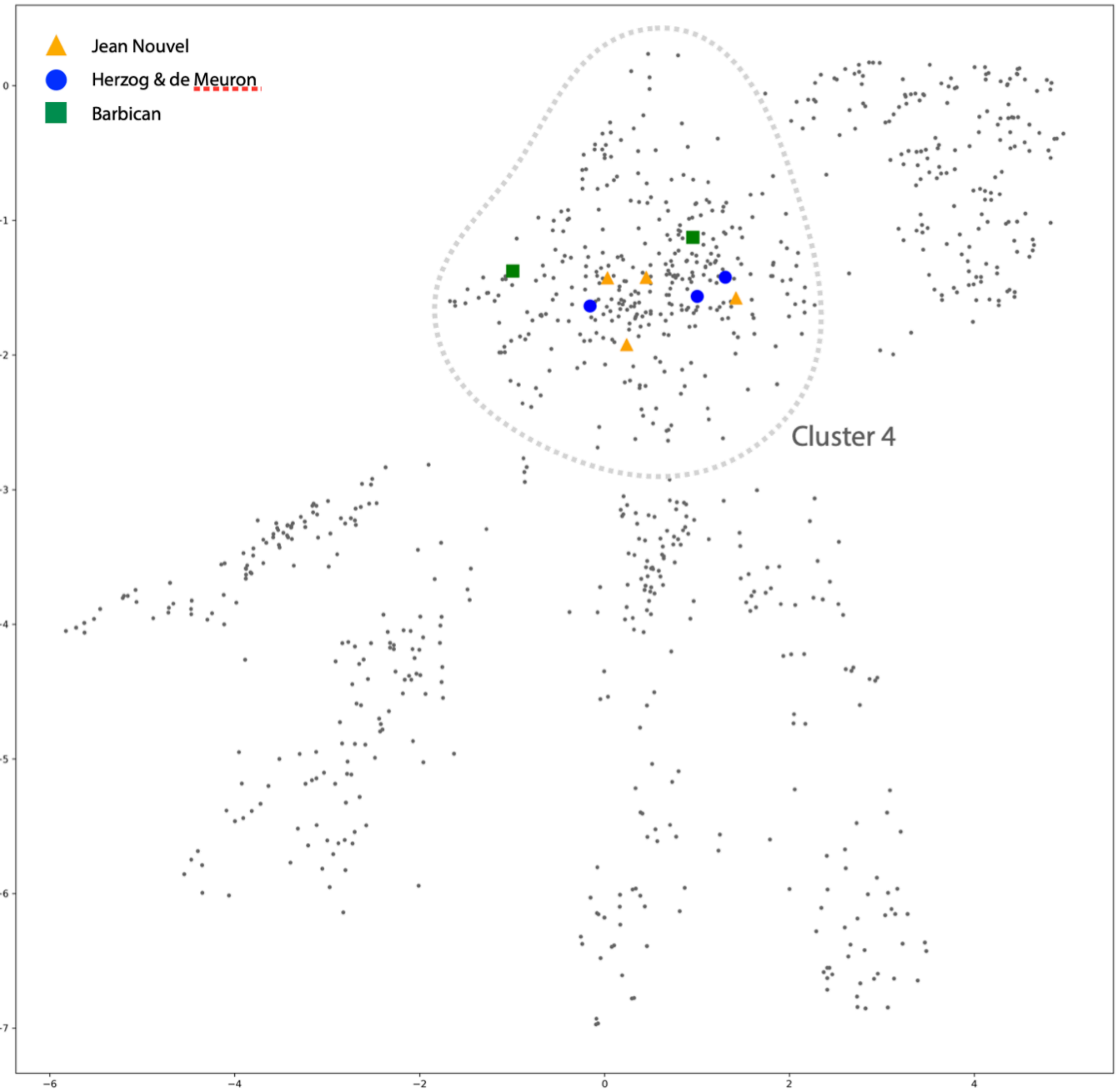


Figure 9: the latent space representation with the dataset and the existing buildings projected on it

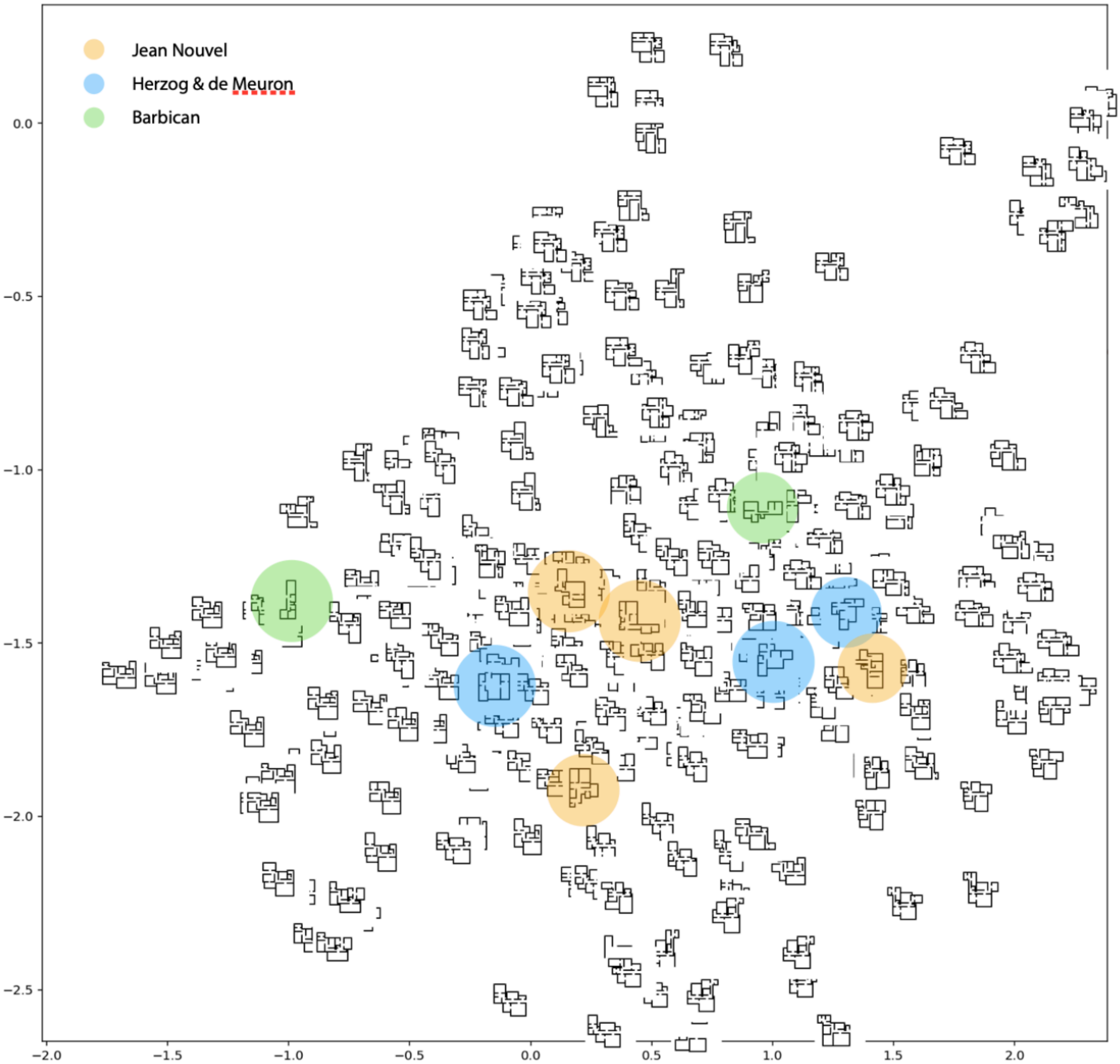


Figure 10: The latent space representation with the dataset and the existing buildings projected on it, zoom on the cluster 4

5. DISCUSSION AND CONCLUSIONS

This research examined a possibility of using ML algorithms as a method for classifying and comparing two-dimensional building floor plans. The methodology consisted of three big stages: creating a dataset of 1000 40 x40 pixel black and white images of apartments floor plans, training of the model with a VAE algorithm, and evaluation of the results with syntactic methods, such as j-graph analysis and mean depth calculation. Moreover, nine real-life floor plans were selected and shown to the algorithm, in order to assess its ability to classify them.

In the course of this study mixed results have been attained. The VAE algorithm outputted a grid which seemed to distribute the floor plans according to an intrinsic logic. First of all, it has been observed that the floor plans which possessed similar shapes tended to be clustered together. Next, the mean depth of each plan type (there has been five plan types identified based on room connections) has been calculated. Mean depth of the whole plan (average of the mean depth values of every room in a plan) has been used in order to check if there is a correlation between the floor plan location on the grid and its mean depth. This correlation has been observed. It could mean that the algorithm has not only identified shape similarities (to some extent) but also the underlying room connection structures.

However, this image distribution could possibly disclose something on what the VAE sees there. As explained in the section 3, the floor plans were generated using an algorithm which had specific rules. For instance if a living room and a kitchen had a common wall, a connection was created. The same goes for the couples bedroom 1 + bathroom 1 and bedroom 2 + bathroom 2. Knowing that the areas of the rooms are predefined, it could be suggested that the number of all possible shapes is limited. Therefore, it is possible that following the logic of the generative algorithm, the deepest floor plan type (type 1 in this case) could only be of a one particular shape variation. The same goes for the other types. By classifying the floor plans according to the shape the VAE can help us determine the rules, by which these shapes were created. In other words, if analysing exclusively the plan distribution, it is possible that one could reestablish the principles by which the generative algorithm was proceeding.

This research attempted to evaluate if the previously generated grid could be used to compare floor plans that are exterior to the initial dataset. It was hypothesised that this grid could be used universally, similarly to some of the human-made building classifications, for instance the "Architectural morphospace" of Phil Steadman, where any rectangular floor plan could be placed (Steadman 2010 and 2014). However, in the case of this study there has been no observed logic in the placement of the new floor plans. There was no shape similarities between the neighbouring plans or any mean depth value correlations. Surprisingly, is some cases the VAE has placed the floor plans from the same building further away from each other that to other building floor plans. Of course that does not mean that there was no logic at all, as the results of machine analysis could not always be understood and interpreted by a human. Moreover, they are not intended to be understood. However, if the aim is to use this classification for comparison, then some degree of understanding is needed, as it would confirm the validity of such comparison.

In summary, the VAE can classify building floor plans according to similarities but it is limited to the training dataset. But that does not mean that the results are not useful. On the contrary, the way in which the algorithm classifies floor plan shapes could gives us some hints about the rules that generated those shapes. In this manner this approach finds its correspondence in shape grammars. These represent algorithmic systems for understanding designs through shape and not through symbols and numbers (Knight 2000). In shape grammars the researchers try to find specific shape rules that help to explain or reproduce a certain type of design (an architect's style for example). It could be said that during this study the ML algorithm could be used to describe the style of the generative algorithm. Thus, it could represent some sort of a prototype of an automatic shape grammar. Another resemblance is that shape grammars are used for generating designs, and VAE has also been used for generative purposes, for instance for creating a street arrangement in the style of London (Varoudis and Penn 2019).

Finally, could this type of task be considered as creative? Surely, the algorithm has outputted a distribution that has never been seen before and could possibly never been produced by a human. On the other hand, when faced with a new type of data, the algorithm could not comprehend it. At this point in a time it is difficult to determine with conviction that this classification is or is not creative. It might not be revolutionary but it could certainly be a step on the way to big discoveries, especially in the field of understanding and reproducing architectural designs. At the moment it could be said that this algorithm represents a tool which could be used to some extent but which should be complemented by human cognition and understanding.

It goes without saying that due to the innovative and experimental nature of this study the process of research encountered several limitations and constraints that were not anticipated. The most challenging part of this type of research becomes gathering a good quality dataset. As it was mentioned in section 3, procurement of a large dataset of building floor plan images could be a particularly time-consuming task, as this type of data, even though available online to some extent, is extremely heterogenous. Thus this type of data is unsuitable for a ML problem. Therefore, the decision was made to generate the dataset automatically, as it allows to obtain a large homogenous dataset considerably faster. However, even in this case time became a constraint. The study used a dataset of 1000 images and it was able to produce results. But it is believed by the author that with more time and a larger dataset (for example 5000 or even 10000 images) these results could have been more precise.

Another limitation of the computer-generated dataset is the generative algorithm itself. Not only the consistency of its functioning could be improved, which is more of a technical issue, but also, as it was seen during the study, it does not generate all the universe of possible architectural form due to how it operates (for example every plan had to have a corridor). It appears that any generative algorithm would have this limitation, as there is always going to be some kind of rules by which the shape arrangement is decided. This is not necessarily a bad thing, as it allows for exploration of the possible architectural form in the given constraints. What is more, it seems to work in synchrony with the VAE, which picks up these rules. However, choosing another dataset generation method for future research could be beneficial. As it was already mentioned, architectural computation and machine learning in architecture is extremely young and is developing massively every year. Since the conduction of this research new methods of floor plan generation have been explored, and these could have been more efficient for a large dataset generation than the method used in this paper (for example, Nauata et al. 2021 and Bisht et al. 2022).

In an ideal world it would be interesting to collect an extensive dataset of real-life building floor plans preprocessed for ML, to conduct a similar type of research and to map these data on a 2D space (which was the original intention of the author before encountering the harsh reality of architectural data quality and availability). Training on the actual real-life data could allow to the algorithm to classify it in relation to each other, which would permit to evaluate the differences between the existing buildings more accurately – a goal that was not entirely achieved during this study (as it was seen, the original dataset, on which the training was performed, was classified in a way that allowed interpretation, but not the consequent adding of existing buildings, which were of different nature and which the ML algorithm did not seem to understand).

Another interesting direction of work continuation in this field is obviously the generative potential of the VAE. As it was seen during the research, and as it was already attempted by previous studies (Varoudis and Penn 2019), the algorithm picks up the generative rules that are behind an architectural style. Possibly, this type of approach could serve as a generative tool for exploring the universe of architectural form or for generating new architectural form in a style of a certain architect for example. It would mean however that a large dataset of that architect's work would also be necessary. That returns us to the problem of data, which is essentially everything in ML.

In conclusion, this research has shown that the usage of ML algorithms within architecture could be promising, but at the same time it might not always return the expected results. This study could be considered as a small contribution to the ongoing exploration and experimentation in this innovative field. More discoveries will surely be made, especially when the problems of lacking or unusable data are resolved and more research is carried out.

REFERENCES

Bisht, S., Shekhawat, K., Upasani, N., Jain, R., Tiwaskar, R., Hebbar, C. (2022). "Transforming an Adjacency Graph into Dimensioned Floorplan Layouts". DOI: 10.1111/cgf.14451.

Benoudjit, M. A., Derix, C., Coates, P. (2004). "Human perception and space classification: The Perceptive Network". *Generative Arts*, Milan.

Buyuklieva, B. (2015). "Machine Learning for Exploring Spatial Affordance Patterns". MSc. University College London.

Chaillou, S. (2019). "AI + Architecture. Towards a New Approach". PhD. Harvard University, available at: https://issuu.com/stanislaschaillou/docs/stanislas_chaillou_thesis_ (accessed on 09/06/2020).

Derix, C., Jagannath, P. (2014). "Digital intuition – Autonomous classifiers for spatial analysis and empirical design". *Journal of Space Syntax*, 5(2), pp. 190-215.

Doersch, C. (2016). "Tutorial on variational autoencoders". arXiv:1606.05908.

Deutsch, R. (2015). *Data-Driven Design and Construction: 25 Strategies for Capturing, Analyzing and Applying Building Data*. Hoboken: Wiley.

Gero, J. S. (1996). "Artificial Intelligence in Computer-aided Design: Progress And Prognosis". *Computer-Aided Design*, 28(3), pp. 153-154.

Goodman, G. (2019). "A Machine Learning Approach to Artificial Floorplan Generation". MSc. University of Kentucky.

Hanson, J. (1998). *Decoding homes and houses*, Cambridge : Cambridge University Press.

Harding, J. (2016). "Dimensionality Reduction for Parametric Design Exploration". DOI: 10.3218/3778-4_19.

Hillier, B. and Hanson, J. (1984). *The Social Logic of Space*, Cambridge: Cambridge University Press.

Knight, T. (2000). "Shape grammars in education and practice: history and prospects". Available at: http://www.mit.edu/~tknight/IJDC/ (accessed on 20/09/2020).

Knight, T., Stiny, G. (2001). "Classical and non-classical computation". *arq: architectural research quarterly*, 5(4), pp.355-372.

Markus. T. A. (1987). "Buildings as classifying devices". *Environment and Planning B: Planning and Design*, 14, pp. 467-484.

Nauata, N., Hosseini, S., Chang, K., Chu, H., Cheng, C., Furukawa, Y. (2021). "House-GAN++: Generative Adversarial Layout Refinement Networks". arXiv:2103.02574v1.

Ostwald, M. J. (2011). "The Mathematics of Spatial Configuration: Revisiting, Revising and Critiquing Justified Plan Graph Theory". *Nexus Network Journal*, 13(2), pp. 445-470.

Rocca, J. (2019). "Understanding Variational Autoencoders (VAEs)". *Towards Data Science*, available at: https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73 (accessed on 27/08/2020).

Roh, Y., Heo, G., Euijong Whang, S. (2018). "A Survey on Data Collection for Machine Learning: A Big Data - AI Integration Perspective". *IEEE Transactions on Knowledge and Data Engineering*, DOI: 10.1109/TKDE.2019.2946162.

Saha, S. (2018). "A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way". *Towards Data Science*, available at: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53 (accessed on 15/09/2020).

Sailer, K. (2018). "Corridors, Classrooms, Classification – The impact of school layout on pedagogy and social behaviours". In Daniels, H. et al, (2018). *Designing for the future of schooling*, Abingdon: Routledge.

Shpuza, E., Peponis, J. (2008). “The effect of floorplate shape upon office layout integration”. *Environment and Planning B: Planning and Design*, 35, pp. 318-336.

Steadman, P., Mitchell, Lj. (2010). “Architectural morphospace: mapping worlds of built forms”. *Environment and Planning B: Planning and Design*, 37(2), pp. 197-220.

Steadman, P. (2014). “Generative Design Methods and the Exploration of Worlds of Formal Possibility”. *Architectural Design*, 84(5), pp. 24-31.

Tamke, M., Zwierzycki, M., Nicholas, P. (2018). “Machine learning for architectural design: Practices and infrastructure”. *International Journal of Architectural Computing*, DOI: 10.1177/1478077118778580

Tarabishy, S., Psarras, S., Kosicki, M., Tsigkari, M. (2020). “Deep learning surrogate models for spatial and visual connectivity”. *International Journal of Architectural Computing*, 18(1), pp. 53-66.

Thirapongphaiboon, T., Hanna, S. (2019). “Spatial distribution of building use: recognition and prediction of use with machine learning”. *12th International Space Syntax Symposium*, Beijing.

Varoudis, T., Penn. A. (2019). “Variational Beauty of Space: Machine Intuition and Non-Linear Neural Aggregations”. *12th International Space Syntax Symposium*, Beijing.

Wu, W., Fu, X., Tang, R., Wang, Y., Qi, Y., Liu, L. (2019). “Data-driven Interior Plan Generation for Residential Buildings”. *ACM Transactions on Graphics*, 38(6), article 234.