Received 14 February 2022
Revised 20 May 2022
29 August 2022
Accepted 14 September 2022

EDU-AI: a twofold machine learning model to support classroom layout generation

Ilker Karadag
Department of Architecture, Faculty of Fine Arts, Design and Architecture, Manisa Celal Bayar University, Manisa, Turkey

Orkan Zeynel Güzelci
Digital Fabrication Laboratory, Faculty of Architecture, University of Porto, Porto, Portugal, and

Sema Alaçam
Department of Architecture, Faculty of Architecture, Istanbul Technical University, Istanbul, Turkey

Abstract

Purpose – This study aims to present a twofold machine learning (ML) model, namely, EDU-AI, and its implementation in educational buildings. The specific focus is on classroom layout design, which is investigated regarding implementation of ML in the early phases of design.

Design/methodology/approach – This study introduces the framework of the EDU-AI, which adopts generative adversarial networks (GAN) architecture and Pix2Pix method. The processes of data collection, data set preparation, training, validation and evaluation for the proposed model are presented. The ML model is trained over two coupled data sets of classroom layouts extracted from a typical school project database of the Ministry of National Education of the Republic of Turkey and validated with foreign classroom boundaries. The generated classroom layouts are objectively evaluated through the structural similarity method (SSIM).

Findings – The implementation of EDU-AI generates classroom layouts despite the use of a small data set. Objective evaluations show that EDU-AI can provide satisfactory outputs for given classroom boundaries regardless of shape complexity (reserved for validation and newly synthesized).

Originality/value – EDU-AI specifically contributes to the automation of classroom layout generation using ML-based algorithms. EDU-AI's two-step framework enables the generation of zoning for any given classroom boundary and furnishing for the previously generated zone. EDU-AI can also be used in the early design phase of school projects in other countries. It can be adapted to the architectural typologies involving footprint, zoning and furnishing relations.

Keywords Artificial intelligence, Machine learning, Generative adversarial networks, Architectural design, Classroom layout, Plan layout generation

Paper type Research paper

Construction Innovation
Vol. 23 No. 4, 2023
pp. 898-914
Emerald Publishing Limited
1471-4175
DOI 10.1108/CI-02-2022-0034

1. Introduction

Social, cultural and economic conditions have been influential in the spatial needs of educational buildings and the general content of education. Similarly, the educational buildings' envelopes, plan layouts and functions have been diversified. Classrooms are commonly accepted as one of the main functions in educational buildings. In classroom design, open-plan layouts became popular in the 1950s because of an increased number of students and a need for adaptability; however, since the 1980s, the cellular organization has remained the most preferred design strategy (Dovey and Fisher, 2014; Shabha, 1993). In classrooms with a cellular plan type, design decisions related to the furniture layout (student desks, board, teacher desk, lockers, etc.) can be decoded according to specific rules. These rules are as follows: the location of the board is chosen according to the position of the door; the teacher's desk is located according to the position of the board; the student desks are arranged according to the position of the board and the teacher's desk; and almost inevitably, the orientation according to the windows.

In the case of Turkey, cellular plan organization of the classrooms is widely used in "typical" school projects. "Typical school projects" refers to the base projects in Turkey adapted to varying levels of education and contexts. In other words, the typical school projects are context-free typologies for school design to be modified according to contextual requirements such as environmental conditions, social needs or level of education.

Regarding the large number of school buildings that have been built and are likely to be built in the future, zoning and furniture relocation in the classrooms can be considered as a repetitive task of selecting a combinatorial solution among a finite number of alternatives. For similar reasons, the decodability of classroom furniture layout makes it easy to be represented via computational methods. Among many computational methods, machine learning (ML) is used in this study. ML algorithms trained with given data sets can manifest a promising performance for school building zoning and furniture relocation problems. This study aims to propose a twofold ML model, namely, EDU-AI, and its implementation for classroom layout generation.

This paper is structured in five sections. After this introduction, Section 2 presents the literature review on ML in architecture, the structure of the Pix2Pix method and studies adopting Pix2Pix for architectural plan layout generation. Then, Section 3 presents the methodology by detailing data collection, data set preparation, two-step training of the model and its validation, followed by results and discussion in Section 4. Last, Section 5 presents the limitations of the model and the potential improvements, and also provides a discussion on the contribution of the study to research and practice.

2. Literature review

2.1 Machine learning in architecture

Negroponte (1975) puts the terms "intelligence" and "understanding" together, indicating the need to address computer graphics and machine vision simultaneously. He defines two approaches: inputting all knowledge to machines in the first step; or alternatively, inputting all knowledge to machines in the first step and imparting the learning process to machines (Negroponte, 1975). Since then, many studies have made it possible to represent architectural knowledge for computers, and demystify the architectural design process, alongside endeavoring to externalize the reasoning process of expert designers. In the 1990s, there were remarkable theoretical contributions in formalizing the design process in a way that computers might become partners. Frameworks for case-based, knowledge-based or data-based design systems (Coyne *et al.*, 1990; Rosenman *et al.*, 1991) have been discussed with a common interest in converting architectural knowledge into parsable tokens and developing expert systems that can interpret existing designs (Gross, 1996).

At present, owing to the diversification of techniques and methods and the available tools for ML, revisiting Negroponte's provisional assumptions and 1990s theoretical frameworks has become possible (Pena *et al.*, 2021; Tamke *et al.*, 2018; Zhang, 2019). Pena *et al.* (2021) provide a comprehensive overview of the potential of computational methods such as generative design approaches (cellular automata, evolutionary computing), artificial neural networks, deep learning and ML used in the earlier phases of design. Tamke *et al.* (2018) put a particular emphasis on the need for customized workflows for ML in the integrated design processes.

Generative adversarial networks (GAN) as a subset of ML techniques was first introduced in 2014 (Goodfellow *et al.*, 2014). There is an increasing interest in applying specific GAN techniques in architectural design processes (Belém *et al.*, 2019; Nauata *et al.*, 2020; Newton, 2019; Zhang, 2019). Belém *et al.* (2019) present a detailed overview of the reflections of ML in the early stages of architectural design. In their review, Belém *et al.* (2019) use a conceptual classification system. Conceptualization, algorithmization, modeling and optimization task approaches are identified. Newton (2019) presents GAN-based techniques related to the existing generative approaches such as optimization, search and probabilistic algorithms and generative grammars in design. GAN have provided a base for the emergence of new tools and techniques, including but not limited to deep convolutional generative adversarial networks (Radford *et al.*, 2015), infoGAN (Chen *et al.*, 2016), Pix2Pix (Isola *et al.*, 2017), StyleGAN (Karras *et al.*, 2019) and House-GAN (Nauata *et al.*, 2020). In brief, advances in ML techniques necessitate the reconsideration of the existing workflows of architectural design processes toward expanding the boundaries of the known.

2.2 Pix2pix

The Pix2Pix (Isola *et al.*, 2017) method is a form of conditional generative adversarial networks (cGAN), in which the output image is generated based on an input source image. The GAN model (Goodfellow *et al.*, 2014) is based on the notion of competition between two sides, "generator" and "discriminator." A generator model for creating new logical synthetic images and a discriminator model for classifying images as real (from the data set) or fake (generated) constitutes the GAN architecture. The generator model is updated via the discriminator model, while the discriminator model is updated directly. As a result, the two models are trained concurrently in an adversarial phase in which the generator tries to deceive the discriminator while the discriminator tries to spot the fake images. Adversarial loss is used to train the generator, which allows it to produce reasonable images in the target domain. Further loss between the generated image and the planned output image is also used to update the generator. The generator model is encouraged to construct reasonable translations of the source image because of the additional loss.

In the case of Pix2Pix, a source image and a target image are given to the discriminator, similar to the GAN architecture, then the discriminator decides whether the target is a reasonable translation of the source image or not. Pix2Pix has been tested on various image-to-image transformation tasks, including translating maps to satellite photos, black-and-white photos to color and product drawings to product photos.

2.3 Pix2Pix for architectural plan layout generation

The adaptation of Pix2Pix in architectural plan layout generation has been investigated by many researchers (Chaillou, 2020; Huang and Zheng, 2018; Liu *et al.*, 2021; Liu *et al.*, 2022; Tian, 2021). Huang and Zheng (2018) adopt Pix2Pix to provide image-to-image translation. They have used Pix2Pix to analyze and generate floor plans of residential buildings. They use "walkway," "bedroom," "living room," "kitchen," "toilet," "dining room," "balcony," "window" and "door" as segmentation labels and by using different color codes they assign labels to the initial data set with a sample size of 115 plans (100 training and 15 validation plan layouts). This model introduced can be executed in two directions: from a plan drawing toward a labeled zoning-like representation and vice versa. Liu *et al.* (2021) implemented Pix2Pix in the site plan scale for campus design. They have achieved reasonable results with a data set including site plans of 85 universities and 302 primary schools. In their study, initial design goals were defined, and it was ensured that the size of the data set fulfills the defined design criteria. They also present the training process results; however, it is not clear from their publication whether the trained code has been tested further with validation data. Chaillou (2020) introduces a onefold plan generation model based on Pix2Pix. The size of the data set used to train the model was more than 800 plan drawings. The first stage consists of zoning information (opening, footprint and entrance) and architectural program (living room, bedroom, closet, kitchen, bathroom and circulation as labels); while in the second turn, the furnishing drawings are included according to the author. The contribution of Tian (2021) is a generic workflow that provides a generative building footprint (color-coded) with a given urban block (site boundary geometry). The study of Tian (2021) can potentially be used in the morphological analysis of urban patterns. Liu *et al.* (2022) focus on adopting Pix2Pix technique to design and analyze the Chinese private garden layouts. The contribution of Liu *et al.* (2022) is related to improving the training process with a small data set (30 gardens) through sequential phases in which the number of labels (7, 9 and 10 labels, respectively, for each training) is increased gradually.

The context, function, scale, representation and size of the data set varies between the studies evaluated (Table 1). Table 1 also demonstrates how workflows using the same method (Pix2Pix) may be customized and structured differently.

3. Methodology

This study proposes a twofold ML model, namely, EDU-AI created to support classroom layout generation by using the Pix2Pix method. The structure of EDU-AI including data collection, preparation of the data set, training of the model and its validation, is presented in this section. The flowchart of the proposed model is displayed in Figure 1.

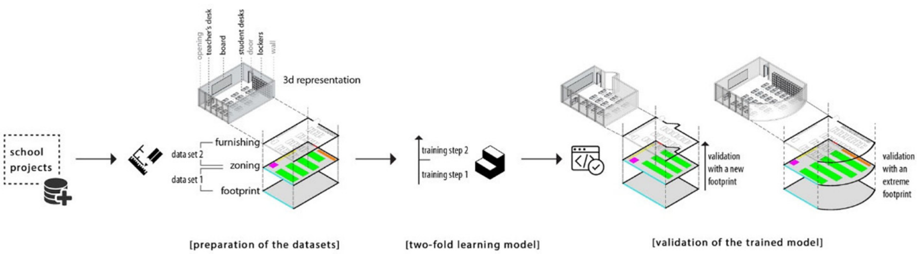
3.1 Data collection

In the scope of the study, the data set is prepared by the authors to train EDU-AI. The data set includes 144 classroom layouts extracted from ten selected typical school projects obtained from the typical school database of the Ministry of National Education of the Republic of Turkey (MEB) (Ministry of National Education of the Republic of Turkey Typical Educational Buildings, 2022). The database of MEB includes typical school projects designed to have 4, 6, 8, 12, 16, 20, 24, 32 or 40 classrooms that can serve all levels of education in Turkey excluding higher education. The ten selected school projects consist of both middle and high schools and have 40 or 32 classrooms. School projects with the highest number of classrooms were selected due to similar classroom types repeating on different floors with varying spatial arrangements. These school projects have a countrywide impact since the ten selected projects from the database have been built repeatedly in different regions of Turkey and are expected to be built again as the need arises. Figure 2 shows floor plans of 6 of 10 selected school projects and the use of colors refers to the differentiation of the classroom types. The differentiation occurs in the proportion of the classroom sizes, number of available student desks, outlines and the spatial relationships among the decoded components (Figure 2).

Study	Context and function	Scale	Structure	Representation	Data set
Huang and Zheng (2018)	Housing	Floor plan	Onefold two directional	Nine colors corresponding to the walkway, bedroom, living room, kitchen, toilet, dining room, balcony, window and door	115 images grouped as 100 training and 15 validation data
Chaillou (2020)	Housing	Floor plan	Onefold one directional	Nine colors corresponding to opening, footprint, entrance, living room, bedroom, closet, kitchen, bathroom and circulation	Around 800 apartment plans as training data and more than 40 validation data
Liu *et al.* (2021)	Educational, campus	Site plan	Onefold one directional	Approximately 20 colors refer to the various components (architectural and landscape) of campuses	387 images (85 university and 302 primary school layouts) as training data. Validation data is not mentioned
Tian (2021)	Urban and regional planning	Site plan	Onefold one directional	16 colors corresponding to different types of functions such as residential and commercial	4400 images grouped as 4000 training and 400 validation data
Liu *et al.* (2022)	Private garden	Site plan	One-fold one directional (iterative)	Ten colors corresponding to boundary, entrance, water, site, central area, pathway, architecture, main landscape architecture, mountain and pavilion	125 images grouped as 120 training and 5 validation data
EDU-AI	Educational, classroom	Single room scale	Twofold one directional	Eight colors corresponding to wall, door, opening, footprint, student desks, teacher desk, board and lockers	162 images grouped as 144 training and 18 validation data

Sources: Huang and Zheng (2018), Chaillou (2020), Liu *et al.* (2021), Tian (2021) and Liu *et al.* (2022)

Table 1. Comparison of the EDU-AI with the studies adopting Pix2Pix in plan layout generation



Sources: Authors

Figure 1. Flowchart indicating the twofold machine learning mechanism of the EDU-AI

3.2 Preparation of the data set

The EDU-AI data set consists of two parts: zoning based on given footprints and arrangement of classroom equipment and furniture based on given zonings. In this study, classroom space and its components are represented by a raster image with eight color

Notes: (a) A middle school with 40 classrooms; (b) a middle school with 32 classrooms; (c) a high school with 40 classrooms; (d) a high school with 32 classrooms; (e) a high school with 40 classrooms; (f) a middle school with 32 classrooms
Sources: Produced by the Authors based on Ministry of National Education of the Republic of Turkey Typical Educational Buildings (2022)

Figure 2. Representation of repeating classroom layouts

codes (Figure 3). The data is organized into two separate training steps. Each step consists of two images:

- Training Step 1: footprint and zoning (1200 × 600 pixels, 96 dots per inch) [Figure 3(a) and 3(b)].
- Training Step 2: zoning and furnishing (1200 × 600 pixels, 96 dots per inch) [Figure 3(b) and 3(c)].

The source and target image representations are first generated in the AutoCAD software. Each building element and furniture type is precisely drawn based on actual sizes. The drawings are rescaled proportionally while converting the vector drawing into a raster image. The authors assigned color codes to the classroom components. However, the model does not directly recognize and use objects (student desks, teacher desk, board, lockers, door and opening) and the spatial relations among them. Therefore, the algorithm's execution is different from a basic conditional replacement operation. In this situation, the algorithm using latent features embedded in the raster image provides rapid solutions based on the trained data sets.

The source and target image representations are extracted from selected typical school project floor plans. Further to the redrawing and conversion to raster images, the classroom

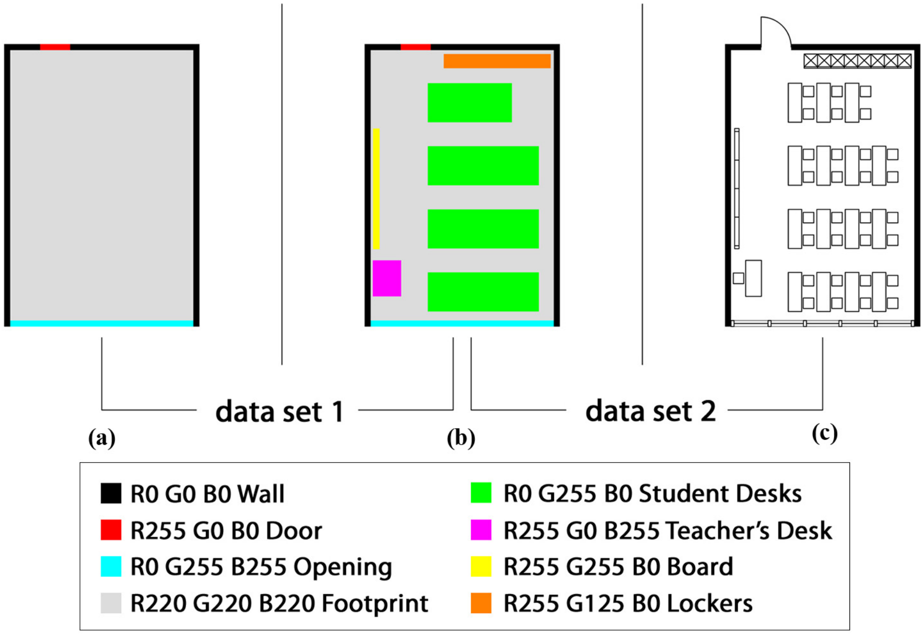


Figure 3.
(a) Footprint; (b) zoning; (c) furnishing

Sources: Authors

layouts are augmented via translation operations such as mirror and rotation. This data augmentation process is performed to achieve better results in the training process. Figure 4 illustrates samples from Data set 1 and Data set 2.

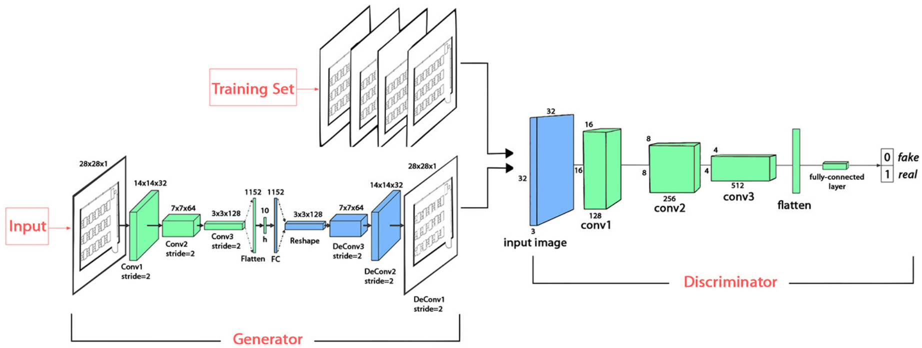
3.3 Training and validation

Further to the data preparation process, the data set is used to train a Pix2Pix GAN model in Keras. Keras is an open-source software library that provides an interface for TensorFlow (an open-source library for numerical computation and large-scale ML) which is used for artificial neural networks. Each image (1200 × 600 pixels) is loaded, rescaled and split into parts titled "footprint plan" and "zoning plan." The output is 144 + 144 color image pairs with a width and height of 256 × 256 pixels. The data set is also split into "train" and "validation" subsets with the k-fold cross-validation method ("k" is taken as 2). Thus, the potential bias in training and validation data sets is overcome. Last, the plan images are enumerated in a directory, loaded each with the target size of 256 × 512 pixels to ensure that the data set is converted to image arrays. Subsequently, image arrays are streamed to the network during training in a format suitable for Pix2Pix. The architecture of the EDU-AI is based on generator and discriminator parts, as shown in (Figure 5).

The generator model is commonly considered an encoder-decoder model. The model takes a source image (e.g. footprint) and generates a target image (e.g. zoned plan). The generator takes source images as input and parses each image into several tokens. The automated encoding-decoding is called convolutional auto-encoder architecture. The generator executes the encoding-decoding process by first down-sampling or encoding the input image down to a bottleneck layer, then up-sampling or decoding the bottleneck representation to the size of the output image. The

Source: Authors

Figure 4. Some samples from Data set 1 and Data set 2



Source: Authors

Figure 5. Generative adversarial neural network architecture

discriminator model takes input from both the training set and generator and determines whether the given images are fake or real.

For the training of GAN models, thousands of iterations are required. An epoch occurs when an entire data set is passed forwards and backwards through the neural network once with a batch size of one. Because the data set of EDU-AI consists of 144 elements, 144 iterations correspond to a single epoch. A batch size of one refers to the entire data set that is not parsed into parts. Gradient descent as an iterative optimization algorithm is used in ML studies to assess the learning process in real time to find the most optimal results. The gradient descent has a parameter called loss function which is correlated with the learning rate. However, the success of the learning process of a GAN model cannot be decided by gradient descent, only; besides, there is also an attempt to establish an equilibrium between the generator and discriminator models. As such, deciding when training should stop is not straightforward.

Many attempts have been made to establish an objective measure of generated image quality. To quantify the overlap percentage between the target and generated image, the structural similarity method (SSIM) is used as a commonly used objective evaluation metric. Simply put, SSIM is correlated with the quality and perception of the human visual system (HVS color model). Therefore, instead of using traditional error summation methods, the SSIM is selected as it models image distortion as a combination of three factors: loss of correlation, luminance distortion and contrast distortion. An SSIM of "0" means that there is no overlap between the images, while an SSIM of "1.00" indicates that the union of the images is the same, as they are entirely overlapping (Wang *et al.*, 2004).

Moreover, in this study, the training models are periodically saved to a *.h5 formatted file during the training (at every 10 training epochs) and used to generate sample image-to-image translations. To achieve this, the generator models are taken at the specified epochs and used to generate translations of three selected empty plans in the data set (explained in detail in the following section). Next, the source, generated images and the target are plotted as three rows of images. Last, to decide on a final model, the generated images are reviewed manually using the saved models.

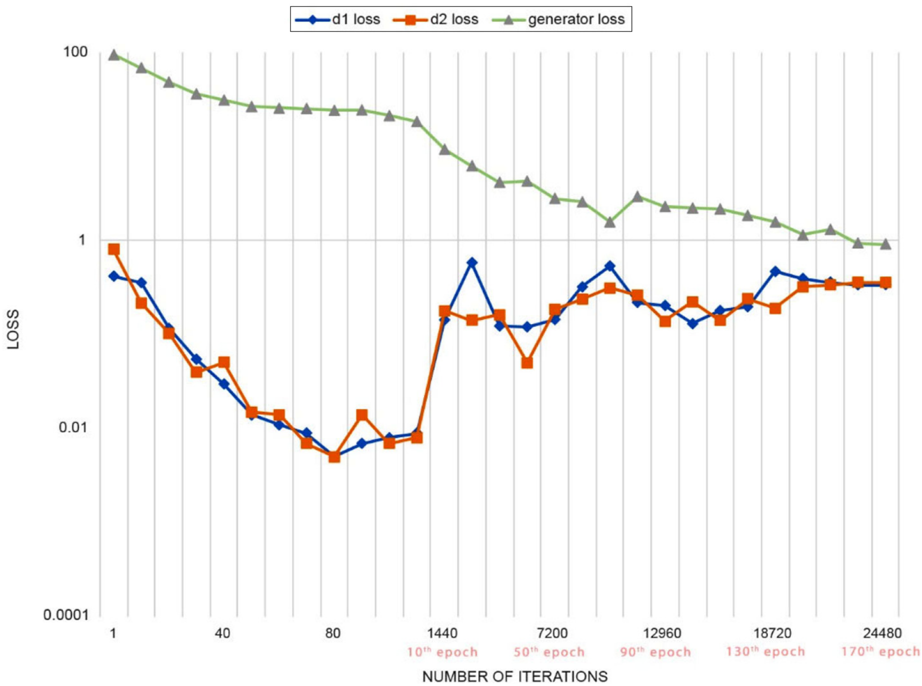
The structure of EDU-AI is based on first using footprints (classroom boundaries) and generating zoning solutions (zones to place furniture) for classrooms. Second, the algorithm uses the outcome of the previous stage as input and generates a classroom plan layout with the furniture as output. To be able to make these sequential operations, Model Step 1 (from footprint to zoning) and Model Step 2 (from zoning to furnishing) are trained separately.

3.3.1 Model Step 1. The equilibrium state in the training process is determined based on two main factors: a numerical result achieved from the adversarial process between discriminator and generator and the empirical results that the user interprets. The loss is calculated at each iteration, including the discriminator loss on real examples (d1 loss), discriminator loss on generated examples (d2 loss) and generator loss, which is a weighted average of adversarial loss (generator loss). In the first training, reasonable outputs began to be seen at iteration of 2,880 (20th epoch), followed then by a horizontal pattern until 7,200. Finally, it achieved a state of equilibrium at 10,080 (70th epoch). Gradient descent started to be seen; however, an iteration number of 24,480 (170th epoch) is the ultimate equilibrium point in which the differences in loss values are not seen. In other words, EDU-AI becomes capable of generating satisfactory results from the training data set after 170 epochs in the first fold. The graph representing the equilibrium state (Figure 6) indicating the loss function values for both discriminator and generator is given below.

In the first training, while there is an intensive learning process in the 10th epoch and the 30th epoch, it is observed that the zoning solutions generated in the 70th and the 170th epoch became more reasonable (Figure 7).

After the training process, Model Step 1 is tested with six footprints randomly chosen from Data set 1. Next, zonings are generated from the plan layouts given in the detail of footprint (exact results) (Figure 8).

Six validation footprints are reproduced from Data set 1 by changing the basic features of footprints, such as the width and length of the floor, besides the



Sources: Authors

Figure 6. Loss values indicating the equilibrium state at the iteration number 24,480 (the graph is logarithmically scaled)

configuration of the walls. These synthesized validation materials are given to Trained Model Step 1 as a source. The yielded SSIM values (varying between 0.71 and 0.97) show that the trained model is performing well on the regular floor plans. On the other hand, in more complicated floor plans, the SSIM value remains within the range of 0.63–0.69 (Figure 9).

3.3.2 Model Step 2. In the second training, reasonable outcomes are seen approximately in the 6,000th iteration and continued until the 8,600th iteration with minor divergences. The authors terminated the training at the 43,200th iteration (300th epoch), after no plausible change was observed in the outcomes (Figure 10).

In the earlier iterations of the training phase (until 70th epoch), the model generated empty classroom layouts. Particularly, while furnishing started to be generated in the 140th epoch, the windows with frames appeared in the 210th epoch. Last, doors containing curved lines started to be generated in the 300th epoch (Figure 11).

Further to completing the training of the model with Data set 2, the Trained Model Step 2 is tested with six zoned plans randomly chosen from Data set 2. The furnished classroom drawings are generated based on the given zoning layouts (Figure 12).

Later, Model Step 2 was run with the results from the validation outputs in Model Step 1. The results of Model Step 2 were successful in validation. Thus, the whole two-stage model was implemented. The SSIM values (varying between 0.76 and 0.83) show that the trained model performed well on the randomly selected floor plans of the validation data set (Figure 13). Last, the trained models (Model Step 1 and Model Step 2)

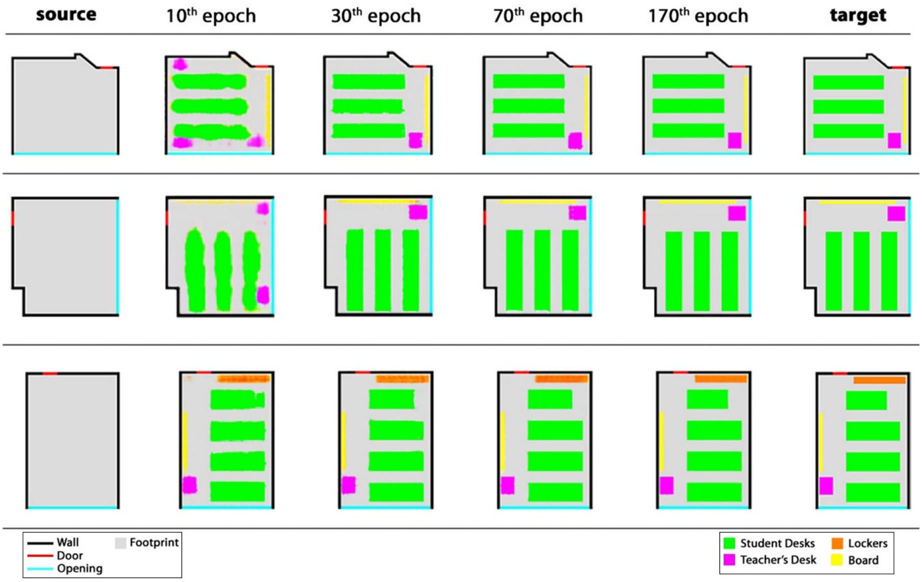


Figure 7.
Learning process of the Model Step 1 with Data set 1 (only the outputs from the critical epochs are given)

Sources: Authors

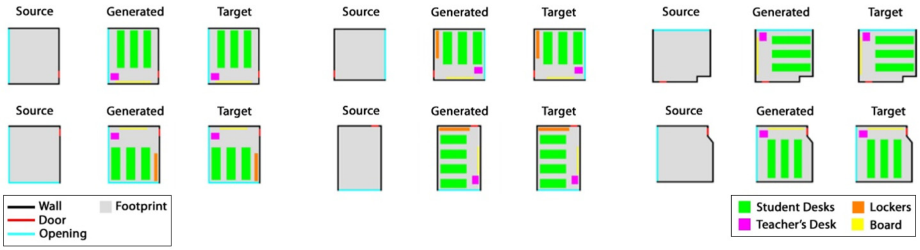
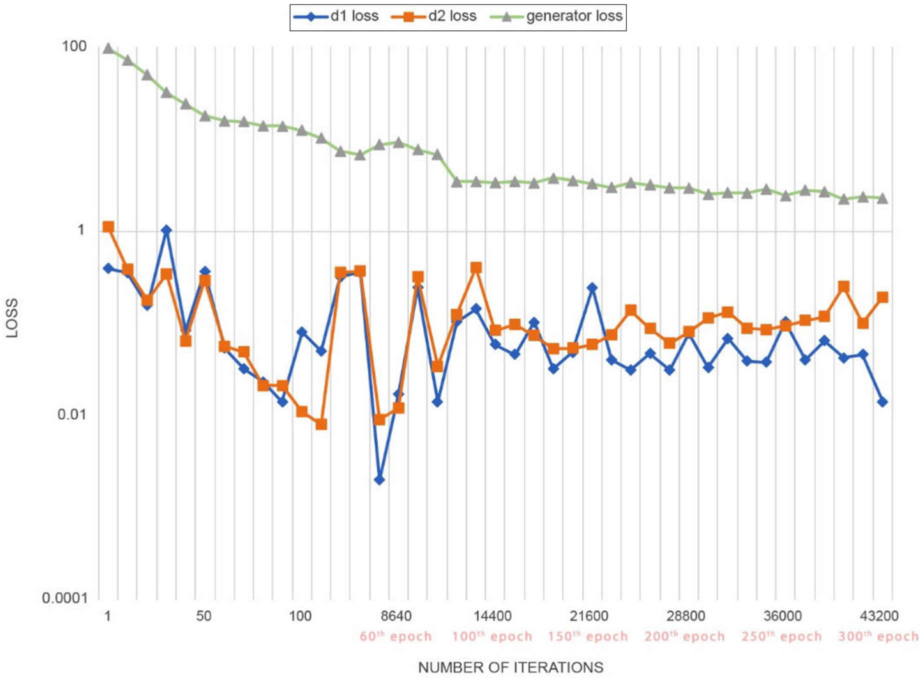


Figure 8.
Test results of the Trained Model Step 1 in randomly chosen source images from Data set 1

Sources: Authors

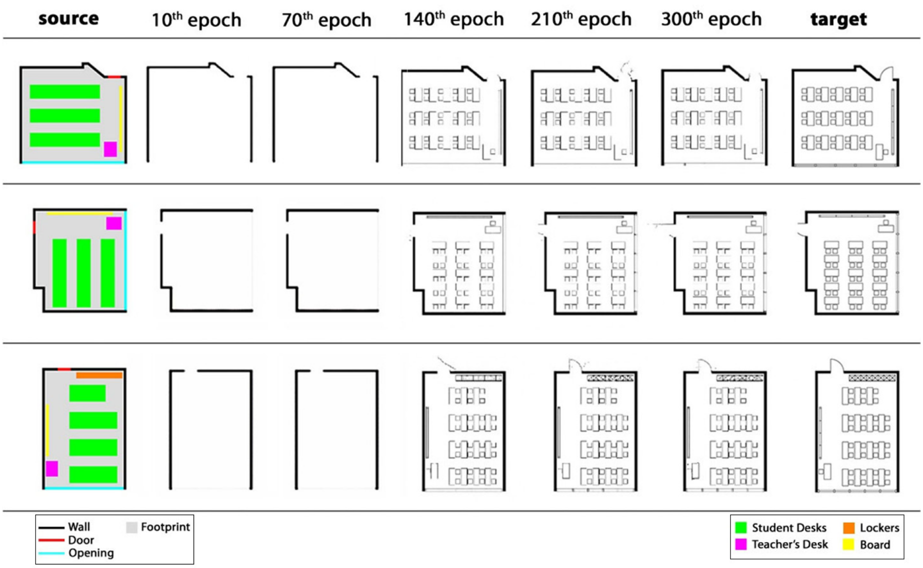
Figure 9.
Test results of the Trained Model Step 1 in validation set 1

Sources: Authors



Sources: Authors

Figure 10. Loss values indicating the equilibrium state at the iteration number 43,200 (the graph is logarithmically scaled)



Sources: Authors

Figure 11. Learning process of the Model Step 2 with Data set 2

created with the proposed twofold learning mechanism were tested with extreme cases (Figure 14).

4. Results and discussion

In contrast to previous studies that used preprepared data for ML problems (Huang and Zheng, 2018), the data in this study is gathered through analogue drawing from architectural precedents. Therefore, the data collection phase of the study is labor intensive. For this reason, the usage of Pix2Pix with a small data set (comparison available in Table 1, "Data set" column) to generate classroom layouts was explored. The outcomes produced

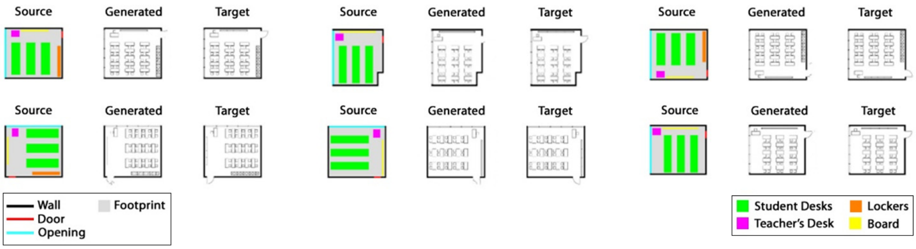


Figure 12. Test results of the Trained Model Step 2 in randomly chosen source images from Data set 2

Sources: Authors

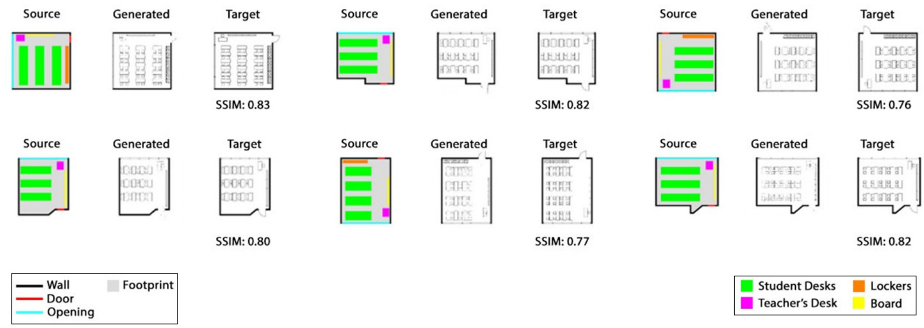


Figure 13. Test results of the Trained Model Step 2 in validation set 2

Sources: Authors

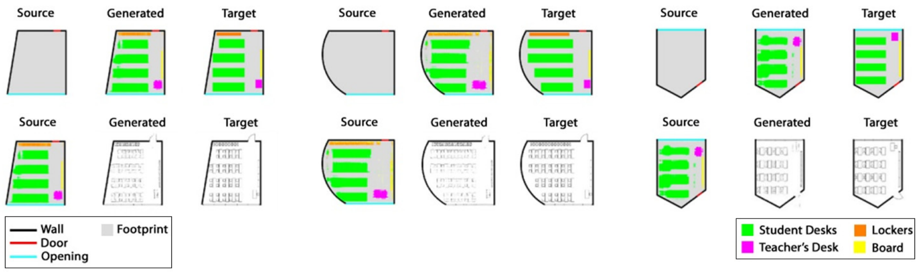


Figure 14. Test results of the trained models ("Model Step 1" and "Model Step 2" created with the proposed twofold learning mechanism) in extreme cases

Sources: Authors

with trained models (Model Step 1 and Model Step 2) indicate that using small data sets for training can easily lead to reasonable outputs. Moreover, augmenting the quality of the ML-based productions can be reconsidered through optimizing the outcomes with the use of advanced computational techniques (e.g. genetic algorithms) and labeling the architectural components and spatial elements.

The proposed model not only succeeded in generating outputs from the training data set but also from the validation data set. In that sense, the results obtained from the validation set reveal the model's potential to be applied to foreign input classes. In the literature, there are lots of samples generated over the training set without indicating whether they are from the training or validation set (Liu *et al.*, 2021). It is considered that generating reasonable outputs for the validation data set provides a degree of flexibility and applicability for other cases involving different complex plan geometries. In other words, the flexibility of ML algorithms is also promising, and the success of the trained model in extreme cases that are not in the training data set shows limitless possibilities. Especially, response of the ML model to the zoning and placing of furnishing within a nonlinear and/or organically shaped boundary shows the flexibility of the model.

One of the crucial findings of the study is that there is no direct correspondence between the shape complexity of the input data (footprint or zoning) and the duration of the processing. This finding indicates that the ML model directly projects what is learnt in the training process onto the generation process without a need for reasoning. Aside from the classroom layout design context in which the ML model was developed, this study raises the question of what other tasks this developed model might contribute to. Considering the footprint–zoning and the zoning–furnishing relationship and translation, the developed model can be applied to floor plans of single function spaces, such as food courts, where the eating zone-circulation area and zoning–furniture relationships are the main issues. Moreover, it can also be implemented with multifunctional typologies such as city halls and hospitals, where the footprint–zoning relationship is spatially established.

Exploring the potentials of a twofold structure can be considered as an initial attempt to generate an interactive ML model that includes multiple tasks, multiple layers and varying levels of complexity in each layer in the context of design iteration. In other words, the proposed ML model deals with atomization of a holistic design process into parts that consist of different types and amounts of data, as well as architectural representation with different levels of abstraction.

The proposed twofold model consists of machine-driven processes, one of which is the input of the other. Different from onefold ML models (Table 1), this study investigates the potentials that a twofold structure might bring. The first ML layer gets a footprint as input and provides zoning, whereas the second ML layer uses zoning information and presents furnishing solutions. It is observed that the second ML layer provides a better performance in comparison with the first ML layer. The generated classroom layouts show that the success of the ML model in the second step (Figure 13) is higher than in the first step (Figure 9). This is quite reasonable as the first step includes a more complicated task in which an empty classroom boundary is used as input, whereas in the second step the ML model has clearer and more tangible input, including color-coded locations for furnishing.

5. Conclusion

In recent years, ML applications in architecture have flourished. Studies on architectural plan layout generation have constituted their domain and the Pix2Pix method has been implemented in various contexts. This study has introduced EDU-AI, a twofold ML model that generates classroom layouts for school projects.

The proposed model has been tested through a classroom layout design problem for educational buildings throughout Turkey. The classrooms are taken as individual spaces. The relationships among different classrooms were not included in the study. Therefore, the proposed ML model currently responds to an individual space – individual function framework. Moreover, the used set of furniture in a classroom space was taken as an initial assumption. In future studies, the complexity of the design problem can be increased and the relations between spaces and functions can be integrated into the model.

The proposed ML model requires designer intervention in the steps such as preprocessing (data preparation), training and data transfer between the steps of the ML model. Therefore, it is not fully automated and the pixel-based final outcomes still need postprocessing to be used as an architectural representation (such as vectoral drawing). Apart from automation of the whole process and generating outcomes using a data set with high resolution, it may be possible to convert raster image outputs to vector drawings using image-sampler tools of parametric design software. While not making an explicit contribution to the existing GAN architecture, this study does present a framework for the use of GAN in a defined design context.

The proposed ML model covers generating design alternatives for the defined context; however, it does not include an optimization process. Additional layers such as agent-based optimizations (visibility–furniture layout relationship, wayfinding and pedestrian flow simulations, risk analysis under emergency states, etc.) or other advanced computational methods (energy-efficiency, passive climatization, etc.) might be integrated to evaluate and optimize outcomes of the ML model in future studies.

Moreover, the spread of advanced computational tools and methods in the field of design may not lead to data-driven methods gaining even more importance. This study proves how publicly available data (typical school projects) can turn into useful models exploiting the ability of ML techniques to extract features and patterns. In that sense, this study can be considered as an experimental and preliminary study that has the potential to provide useful insight in the context of workflow in the production of design alternatives for different details and layers from the data by using ML models.

The use of EDU-AI can provide a design idea that may be helpful in the early design phase not only for architects, but also other stakeholders who are involved in the decision-making process. This type of ML model may help architects save time spent on repetitive tasks and compare their designs (subjective) with the generated outcomes of the trained model based on a data set (objective). In this way, architects using the ML model can use their expertise and knowledge to solve ill-defined problems. In addition, EDU-AI may be used as a decision-support tool for other nonexpert actors.

References

Belém, C., Santos, L. and Leitão, A. (2019), "On the impact of machine learning: architecture without architects?", *18th International Conference, CAAD Futures 2019, Daejeon, Korea*, pp. 148-167.

Chaillou, S. (2020), "ArchiGAN: artificial intelligence x architecture", in Yuan, P.F., Xie, M., Leach, N., Yao, J. and Wang, X. (Eds), *Architectural Intelligence*, Springer, Singapore, pp. 117-127.

Chen, X., Duan, Y., Houthooft, R., Schulman, J., Sutskever, I. and Abbeel, P. (2016), "InfoGAN: interpretable representation learning", *Thirtieth Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain*, pp. 2172-2180.

Coyne, R.D., Rosenman, M.A. and Radford, A.D. (1990), *Knowledge-Based Design Systems*, Addison-Wesley, Reading, MA.

Dovey, K. and Fisher, K. (2014), "Designing for adaptation: the school as socio-spatial assemblage", *The Journal of Architecture*, Vol. 19 No. 1, pp. 43-63.

Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. and Bengio, Y. (2014), "Generative adversarial nets", *27th International Conference on Neural Information Processing Systems – (NIPS), Montreal, Canada*, pp. 2672-2680.

Gross, M.D. (1996), "The electronic cocktail napkin – a computational environment for working with design diagrams", *Design Studies*, Vol. 17 No. 1, pp. 53-69.

Huang, W. and Zheng, H. (2018), "Architectural drawings recognition and generation through machine learning", *38th Annual Conference of the Association for Computer-Aided Design in Architecture, Mexico City, Mexico*, pp. 156-165.

Isola, P., Zhu, J.Z., Zhou, T. and Efros, A.A. (2017), "Image-to-image translation with conditional adversarial networks", *Thirtieth IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI*, pp. 1125-1134.

Karras, T., Laine, S. and Aila, T. (2019), "A style-based generator architecture for generative adversarial networks", *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA*, pp. 4396-4405.

Liu, Y., Luo, Y., Deng, Q. and Zhou, X. (2021), "Exploration of campus layout based on generative adversarial network", *2020 DigitalFUTURES: The 2nd International Conference on Computational Design and Robotic Fabrication, Shanghai, China*, pp. 169-178.

Liu, Y., Fang, C., Yang, Z., Wang, X., Zhou, Z., Deng, Q. and Liang, L. (2022), "Exploration on machine learning layout generation of Chinese private garden in Southern Yangtze", *The International Conference on Computational Design and Robotic Fabrication, Singapore*, pp. 35-44.

Ministry of National Education of the Republic of Turkey Typical Educational Buildings (2022), "Milli eğitim bakanlığı eğitim yapıları tip projeleri", available at: https://tipproje.meb.gov.tr/?doing_wp_cron=1644681827.2129850387573242187500 (accessed 29 August 2022).

Nauata, N., Chang, K.H., Cheng, C.Y., Mori, G. and Furukawa, Y. (2020), "House-gan: relational generative adversarial networks for graph-constrained house layout generation", in Vedaldi, A., Bischof, H., Brox, T. and Frahm, J.M. (Eds), *Computer Vision – ECCV 2020*, ECCV 2020. Lecture Notes in Computer Science, Springer, Cham, pp. 162-177.

Negroponte, N. (1975), *Soft Architecture Machines*, MIT Press, Cambridge, MA.

Newton, D. (2019), "Generative deep learning in architectural design", *Technology| Architecture + Design*, Vol. 3 No. 2, pp. 176-189.

Pena, M.L.C., Carballal, A., Rodríguez-Fernández, N., Santos, I. and Romero, J. (2021), "Artificial intelligence applied to conceptual design. A review of its use in architecture", *Automation in Construction*, Vol. 124, p. 103550.

Radford, A., Metz, L. and Chintala, S. (2015), "Unsupervised representation learning with deep convolutional generative adversarial networks", *Fourth International Conference on Learning Representations, San Juan, Puerto Rico*, pp. 1-16.

Rosenman, M.A., Gero, J.S. and Oxman, R.E. (1991), "What's in a case: the use of case bases, knowledge bases, and databases in design", *CAAD Futures '91 Computer Aided Architectural Design – Education, Research, Application*, Vieweg, Wiesbaden, pp. 285-300.

Shabha, G.S. (1993), "Flexibility and the design for change in school buildings", *Architectural Science Review*, Vol. 36 No. 2, pp. 87-96.

Tamke, M., Nicholas, P. and Zwierzycki, M. (2018), "Machine learning for architectural design: practices and infrastructure", *International Journal of Architectural Computing*, Vol. 16 No. 2, pp. 123-143.

Tian, R. (2021), "Suggestive site planning with conditional GAN and urban GIS data", *2020 DigitalFUTURES: The 2nd International Conference on Computational Design and Robotic Fabrication*, *Shanghai, China*, pp. 103-113.

Wang, Z., Bovik, A.C., Sheikh, H.R. and Simoncelli, E.P. (2004), "Image quality assessment: from error visibility to structural similarity", *IEEE Transactions on Image Processing*, Vol. 13 No. 4, pp. 600-612.

Zhang, H. (2019), "3D model generation on architectural plan and section training through machine learning", *Technologies*, Vol. 7 No. 4, p. 82.

Corresponding author
Orkan Zeynel Güzelci can be contacted at: oguzelci@arq.up.pt