Machine learning for conservation of architectural heritage

İlker Karadag
Manisa Celal Bayar University, Manisa, Turkey

Received 13 May 2022
Revised 2 August 2022
2 September 2022
Accepted 5 September 2022

Abstract

Purpose – Accurate documentation of damaged or destroyed historical buildings to protect cultural heritage has been on the agenda of architecture for many years. In that sense, this study uses machine learning (ML) to predict missing/damaged parts of historical buildings within the scope of early ottoman tombs.

Design/methodology/approach – This study uses conditional generative adversarial networks (cGANs), a subset of ML to predict missing/damaged parts of historical buildings within the scope of early Ottoman tombs. This paper discusses that using GAN as a ML framework is an efficient method for estimating missing/damaged parts of historical buildings. The study uses the plan drawings of nearly 200 historical buildings, which were prepared one by one as a data set for the ML process.

Findings – The study contributes to the field by (1) generating a mixed methodological framework, (2) validating the effectiveness of the proposed framework in the restitution of historical buildings and (3) assessing the contextual dependency of the generated data. The paper provides insights into how ML can be used in the conservation of architectural heritage. It suggests that using a comprehensive data set in the process can be highly effective in getting successful results. The findings of the research will be a reference for new studies on the conservation of cultural heritage with ML and will make a significant contribution to the literature.

Research limitations/implications – A reliable outcome has been obtained concerning the interpretation of documented data and the generation of missing data at the macro level. The framework is remarkably effective when it comes to the identification and re-generation of missing architectural components like walls, domes, windows, doors, etc. on a macro level without details. On the other hand, the proposed methodological framework is not ready for advanced steps of restitution since every case of architectural heritage is very detailed and unique. Therefore, the proposed framework for re-generation of missing components of heritage buildings is limited by the basic geometrical form which means the architectural details of the mentioned components including ornaments, materials, identification of construction layers, etc. are not covered.

Originality/value – The generic literature as to ML models used in architecture mostly constitutes design exploration and floor plan/urban layout generation. More specific studies in the conservation of architectural heritage by using ML mostly focus on architectural component recognition over 3D point cloud data (1) or superficial damage detection of heritage buildings (2). However, we propose a mixed methodological framework for the interpretation of documented architectural data and the regeneration of missing parts of historical buildings. In addition, the methodology and the results of this paper constitute a guide for further research on ML and consequently contribute to architects in the early phases of restitution.

Keywords Machine learning, Advanced computational methods, Restitution, Conservation of architectural heritage, Early ottoman period tombs

Paper type Research paper

1. Introduction

The conservation of cultural heritage has long involved accurate documenting of damaged or destroyed historical buildings. Conservation of cultural heritage consists of several stages including documentation, analysis studies and restitution/restoration projects. Restitution studies should be carried out systematically to preserve the buildings in their original form. Examining buildings from comparable periods to gain an early concept of the architectural characteristics of the building is one of the most effective methods employed, especially in the context of damaged/destroyed buildings. This method, on the other hand, may considerably prolong the restitution process as it may require a detailed analysis of many buildings. However, the use of machine learning (ML) techniques has the potential to support architects in this process, especially in the preliminary stages of restitution. In this regard, the aim is to

Open House International
Vol. 48 No. 1, 2023
pp. 23-37

0168-2601
DOI 10.1108/OHI-05-2022-0124

investigate the usage and effectiveness of ML in both the assessment of missing or damaged elements of early Ottoman tombs and the detection of dome typologies.

The use of ML in architecture started with the spread of advanced computational design methods such as rule-based systems, generative systems, and knowledge-based systems. The common ground of computational design in architecture is formed by formulating the design process through design rules and parameters (1), automating repetitive tasks (2), and supporting designers' decisions (3) (Bollinger *et al.*, 2010; Duarte, 2005; Herr and Thomas, 2007; Turrin *et al.*, 2011).

Advanced computational design methods provide new tools to architects; however, ML techniques offer some other possibilities to architects. In 1975, Nicholas Negroponte published a book entitled "Soft Architecture Machines" to encourage the use of ML in the design process. The author combines the terms "intelligence" and "understanding" to underline the significance of considering both computer graphics and machine vision at once. He defines the two-step approach as follows: providing data to the machines in the first step and initiating the ML process from this data in the second step. This approach has also enabled many studies aiming to represent architectural data on computers. In the 1990s, theoretical studies commonly addressed creating a system in which the design was represented with parameters so that computers could also be part of the process. Frameworks for case-based, knowledge-based, or data-based design systems have been tried to be drawn (Coyne *et al.*, 1990; Rosenman *et al.*, 1991). The development of advanced tools that can analyze architectural designs and break down architectural information into decomposable sub-components has become an objective (Gross, 1996; Chouchoulas and Day, 2007; Erem and Ermiyagil, 2016).

Today, developments in ML techniques expand the boundaries of the known and enable the discussion of reconsidering architectural design processes. When it comes to creating architectural plans or furnishing spaces, ML techniques offer new multidimensional possibilities to reformulate and automate design processes. Artificial neural networks have started to be of use especially in the performance analysis of decision support systems, creating prediction models, and solving various design problems. Following the early models, generative adversarial networks (GANs), a subset of ML methods, were released in 2014 (Goodfellow *et al.*, 2014), and they served as a significant basis for the development of several new tools and techniques. DCGAN (Radford *et al.*, 2015), iGAN/GVM (Zhu *et al.*, 2016), infoGAN (Chen *et al.*, 2016), Pix2Pix (Isola *et al.*, 2017) and ArchiGAN (Chaillou, 2020 – uses Pix2Pix) can be counted among these tools. A GAN is an architecture for training deep learning-based generative models. The GAN architecture has been tested in a variety of tasks such as "Image-to-Image Translation with Conditional Adversarial Networks". Converting maps to satellite photographs, black-and-white photos to color, daylight photos to nighttime photos and product drawings to product photos are examples of image-to-image tasks.

The effectiveness of GAN in "Image-to-Image Translation" tasks, especially in the generation of architectural plans, has been investigated by many researchers. Floor plan generation using GAN was first studied by Huang and Zheng (2018). The authors tried to create/identify floor plans using the Pix2PixHD GAN model. They were able to transform the floor plans into color-coded architectural zones. They were also able to reverse-engineer these zones into furnished floor plans. The trained model obtained from the ML process, asks the user for an empty plan with the location of the windows and rooms as input and can produce furnished floor plans for each space as an output. Many similar studies followed, using ML techniques to create floor plans or place furniture in empty architectural spaces (Newton, 2019; Zhang, 2019; Chaillou, 2020; Liu *et al.*, 2021; Nauata *et al.*, 2020).

The reported generic literature review indicates that ML techniques are mostly used in the field of architecture to support or automate the design process. A more specific literature review was also conducted to provide an overview of current research using ML in the field of conservation of architectural heritage.

Most of the studies are focusing on the diagnosis, analysis and classification of the structural components of historical buildings like domes, columns, vaults, *etc.* (Obeso *et al.*, 2016; Llamas *et al.*, 2017; Bassier *et al.*, 2017; Grilli and Remondino, 2020; Croce *et al.*, 2021; Pepe *et al.*, 2022). Thus, aiming to contribute to the conservation and restoration by extracting valuable knowledge from the heritage images. Llamas *et al.* (2017) used convolutional neural networks (CNN), a form of ML model, to classify over 10,000 photos (predominantly churches and religious temples) into 10 types of architectural components. Obeso *et al.* (2016) have labeled 16,000 images in four categories, out of which three are Mexican buildings (pre-Hispanic, colonial, modern) and one "other" by also using CNN. Grilli and Remondino (2020) have classified 3D point cloud data into sub-architectural components like floor, façade, column, arch, vault, window, *etc*. Pepe *et al.* (2022) have also conducted a point cloud classification study in a cultural heritage environment. The authors identified the building components (lintel, capital, column, *etc.*) from point cloud data by also using ML techniques.

ML is also commonly used to assess the structural health condition of architectural heritage buildings to prolong their remaining service life by predicting locations of superficial damages on the surface of the building due to weathering effects, material loss, efflorescence, seepage, algae growth, and moss deposition, *etc.* (Valero *et al.*, 2019; Wang *et al.*, 2019; Mishra, 2021; Loverdos and Sarhosis, 2022).

The aim of this study, in contrast to other studies, is to employ ML to assist architects in a new field named "the preliminary stages of restitution." The use of ML in restitution is expected to contribute to the opening of new perspectives in the field. A critical research problem for the conservation of architectural heritage is trying to be solved by a novel approach which is not an experimental attempt but a confident step forward to be of use in the preliminary steps of restitution by architects. However, because each case of architectural heritage is unique and sensitive, it is not claimed that the offered method can be used directly in the advanced steps of restitution. However, this study provides a "new macro level mixed methodological framework" that is thought to be effective for both interpreting documented data and generating missing data in the early stages of restitution.

2. Material and method

The framework of the proposed model, the methods used, the preparation of the data set, and the training process of the model are all presented in this section. Figure 1 illustrates the flow chart of the study.

A mixed methodological framework is used in this study which includes both interpretation of the dataset (1) and re-generation of new data over it (2). The material sub-chapter (2.1) covers the first part of the proposed framework in which interpretation of the building heritage data is conducted. The method sub-chapter (2.2) contains the details of the

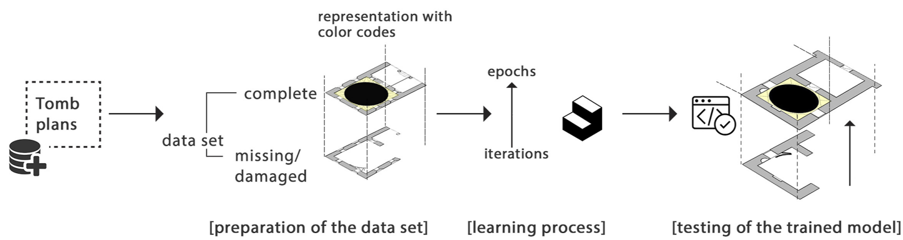


Figure 1.
The flow chart of the study

proposed framework's second part, in which different ML models are investigated for use in the regeneration of missing components of heritage buildings using incomplete input data.

2.1 Material

The material of the study consists of 192 plan drawings produced from the early Ottoman tombs. The origins of the tomb typology in Islamic Architecture date back to the second half of the 9th century in which the first tomb was constructed for an Abbasi caliph. After the 10th century, the tombs started to be seen in Anatolia during the Seljuk period. Then, between the 13th and 15th centuries, early Ottoman tombs began to be seen.

In general, early Ottoman tombs have mostly a square plan with an entrance extension. The plan dimensions range from "5 m × 5 m" to "13 m × 13 m". The height of these tombs is mostly around 10 m, but a few examples are reaching 18 m in height. Entrance openings, mihrab and arched windows are architectural components of early Ottoman tombs, while masonry walls, transition elements to the dome, and the dome as a roof cover element are structural components. For these tombs, there exist three different transitions from the square plan to the dome: (1) pendentive, (2) tromp and finally (3) Turkish triangle. The mechanical/structural relationship between the plane of an Ottoman Tomb and the three transition to dome types is indicated in Figure 2.

For the first part of the proposed framework, two separate data sets are required: the incomplete plan of tombs partly missing many architectural components like domes, walls, windows, entrance, *etc*. (1), and the original complete versions of the tombs (2) (Figure 3a). To prepare the first data set, conscious reductions were made from the full/complete plans of the tombs. The dome, as the main structural component of the Ottoman Tombs, was important in this process because it provides the tomb's structural integrity. For this aim, many studies on the responses of traditional masonry domes to external forces are researched. These domes' vulnerability to seismic and environmental impacts is noticeable (Kuban, 1987; Turan, 1993; Karaesmen, 1993). The lateral loads due to earthquakes are one of the primary factors leading to the collapse of the dome structure (Atamturktur and Sevim, 2012; Ozturk *et al.*, 2019). The vulnerability to environmental effects increases on the remaining structure after the collapse of the dome. The masonry walls lose the structural integrity provided by the dome before and begin the collapse. This was the main criterion when preparing the missing/damaged versions of the tomb plans.

After preparing the first data set, a second data set which constitutes plan drawings of original complete versions of tombs (Figure 3b) is colored (Figure 3c) to use in the training process. Using color codes in image-to-image translation tasks is a widespread practice (Newton, 2019; Zhang, 2019; Chaillou, 2020; Liu *et al.*, 2021; Nauata *et al.*, 2020). In particular, the representation of different components with contrasting colors accelerates the learning process and increases the success rate (Isola *et al.*, 2017; Liu *et al.*, 2021). The colors used in this

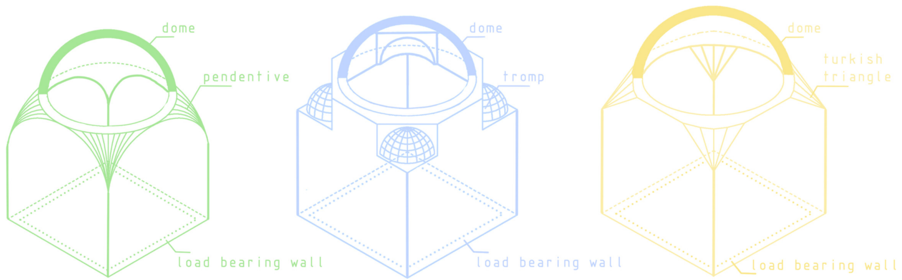


Figure 2.
Plane to dome transition types in Ottoman Tombs

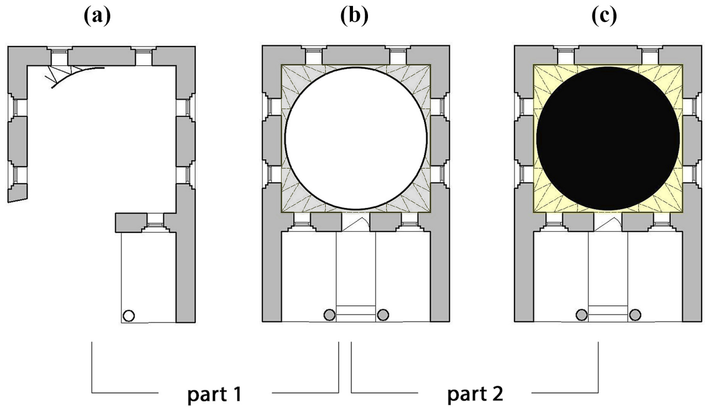


Figure 3.
(a) Missing/damaged; (b) original drawing; (c) complete (represented with color codes)

study were chosen after examining color codes, which are frequently used in ML to help the trained model identify different transitions to dome types (see Figure 4).

The vector drawings prepared from the architectural plans of the 192 Ottoman tombs are pixelated to be of use as input in the ML process. The train-test split method is used to assess the performance of the trained model. This method can be used for any supervised learning technique requiring classification or regression tasks. The main procedure of this method is to divide a data set into two subsets ("training" and "test") using random selection. This is to ensure that the train and test datasets are representative of the original dataset. Thus, the potential bias in "train" and "test" data sets is overcome. The training dataset is used to fit the model during the learning process. The "test data set" is not used during the learning process; instead, after the learning process is complete, the trained model is given random inputs from the test subset. Thus, the trained model's performance is measured on new/unseen data that was not used in the learning process. The size of the train and test sets is the method's key configuration parameter. In our study, a training set with a size of 0.80 (80%) is given to the learning process, while the remaining percentage of 0.20 (20%) is used for testing. There is no optimal split percentage given in the literature (Newton, 2019; Zhang, 2019; Chaillou, 2020; Liu *et al.*, 2021; Nauata *et al.*, 2020) since these values are unique for every ML study. But, in general, they are defined to accurately represent both the training and test process.

The data enrichment/augmentation strategy, which is also widely seen in the literature, has been utilized to enhance the learning process (Huang and Zheng, 2018; Chaillou, 2020). For this, every plan was duplicated by rotating and mirroring in multiples of 90°, bringing the total quantity of data to "1098." This method is beneficial and necessary since every new variation is different in pixels from the perspective of the GAN model. However, identical variations are also created due to the mirroring when the plan of a tomb is symmetrical. In such a case, the duplication is omitted from the data set, manually.

2.2 Method

The second part of the proposed framework including the re-generation of missing architectural components of tombs is given in this sub-chapter. GANs are researched in detail before setting up the framework since image-to-image translation constitutes an important part of the proposed method. The GAN model as a class of ML frameworks was first developed by Goodfellow *et al.* (2014). The term GAN can be described in three separate parts: "generative" describes how data are generated visually (1), "adversarial" means the training of the model is done in an adversarial setting (2), and "networks" imply deep neural networks

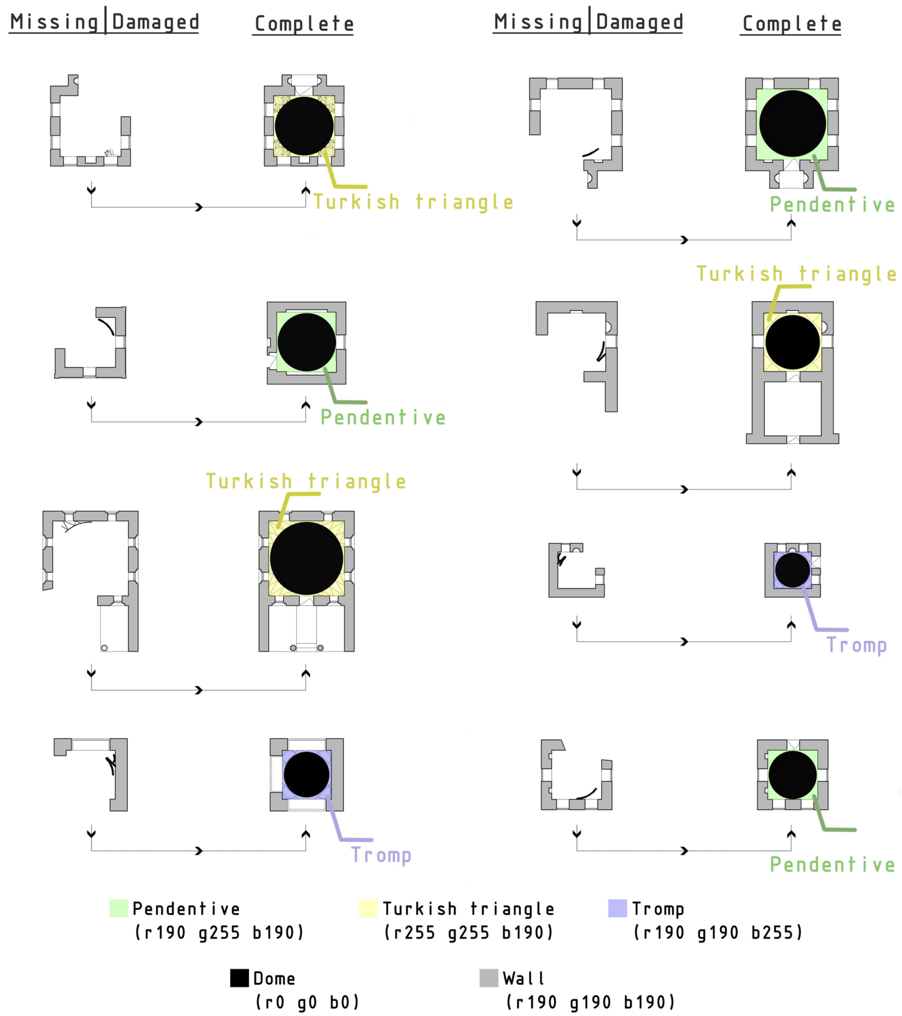


Figure 4.
Color-coded samples from the data set

are used for training purposes (3). The GAN model is based on the concept of rivalry between two networks called "Generator" and "Discriminator". The discriminator tries to classify the distinctive images as real (from the data set) or fake (created by the generator). On the other hand, the generator tries to construct new reasonable artificial images to deceive the discriminator (Goodfellow *et al.* 2014). The system is named GANs since it is formed by two competitive networks (Figure 5).

The original GAN model aims to randomly generate realistic output. However, in this study, to reach the complete plan of a tomb by using the missing/damaged tomb plan as an input which implies a predefined condition is aimed. Therefore, a conditional image-to-image translation task is required which is available with conditional generative adversarial network (cGAN). In cGANs, both the generator and discriminator are conditioned on some extra information which could be any auxiliary data (Mirza and Osindero, 2014).

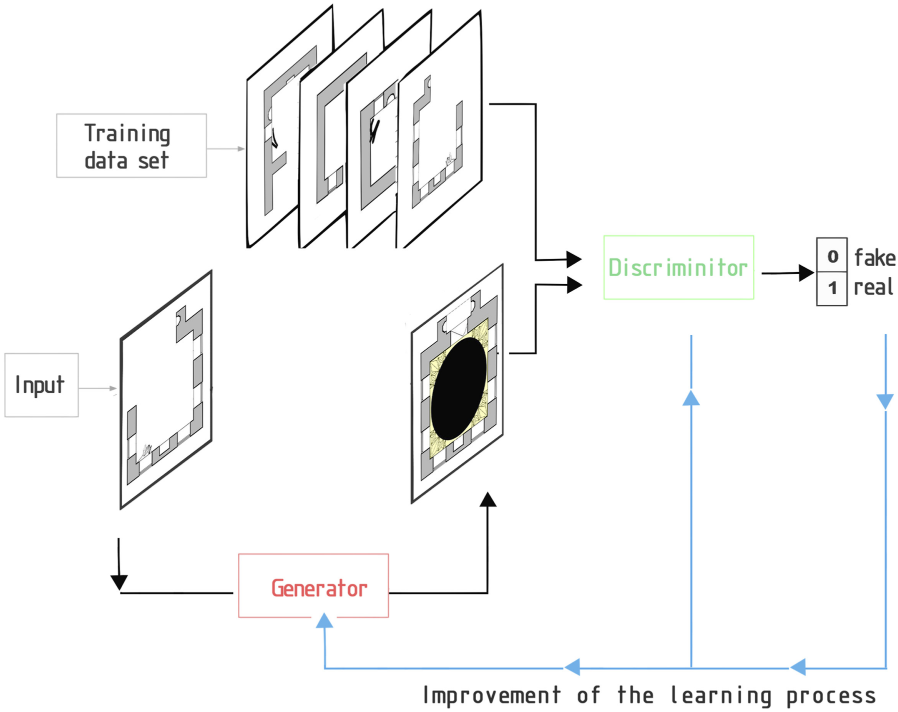


Figure 5.
The architecture of the generative adversarial network (GAN) of the study

As a widely used cGAN model, Pix2Pix has enabled a variety of applications, but the results are often limited to low resolution and unrealistic. On the other hand, an improved version of Pix2Pix named Pix2PixHD is similar to Pix2Pix in a general frame but can provide high-resolution output. The image-to-image translation part of the proposed framework in which the missing architectural components are tried to be predicted can be conducted by utilizing this ML model. Moreover, when it comes to architectural plan drawings, high resolution can provide a significant advantage. Because drawings are converted into image arrays to be of use in the ML process.

In Pix2PixHD, the input and target data sets are required to be organized in two separate folders. Therefore, the incomplete versions of the plan drawings (input) and the colored versions of the original plans (target) are exported separately from the Computer Aided Design (CAD) environment as 1024 × 1024 image arrays. When naming the image arrays, it is necessary to ensure that, for example, if the incomplete/damaged version of the "x" tomb plan was named "x plan.jpeg" in the "input" folder, the complete version in the "target" folder should have the same file name. Thus, two distinct training folders ("input" and "target") with correspondingly "paired images" are provided for the learning process.

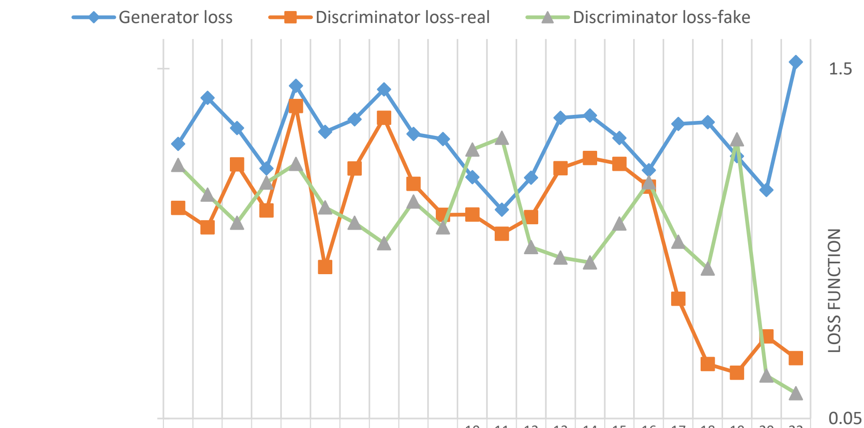
The open-source software library Keras was chosen to train the model on the provided data set because it provides a straightforward and effective Python API for accessing TensorFlow, another open-source library for numerical computation and large-scale ML. Keras provides an interface to TensorFlow for building artificial neural networks, while TensorFlow offers full Keras integration, making advanced ML easier and more convenient (Gulli *et al.*, 2019).

The training process started with the data set brought into the appropriate format. Typically, in the case of GAN models, thousands of iterations are required to reach an equilibrium/final state. Gradient descent was therefore used as an iterative optimization algorithm within the scope of the study to evaluate the ML process in real time and to reach the most accurate outputs. The gradient descent has a parameter called the loss function, which is related to the learning rate. While the generator has a total loss function (generator loss), the discriminator has two loss functions since it has two separate labels as both real (discriminator loss-real) and fake (discriminator loss-fake). While the generator tries to keep its loss function at the maximum, the discriminator tries to keep it at the minimum level. However, the success of a ML process, particularly in the field of architecture, may not be directly decided by gradient descent; because it is a mathematical value and may not always match the user's intended output. Therefore, in addition to this, throughout the learning process, models of all epochs were saved and each of them was manually reviewed to decide on a final model. Moreover, several objective evaluation metrics which are commonly preferred to quantify the similarity between the target image and the generated one were compared, and a decision was given (the details of the decision process are given in the next chapter).

3. Analysis and results

The ML process was continued until the equilibrium state was reached. In this context, both the gradient descent graph and the image outputs were reviewed. The first meaningful outputs were obtained in the 60th cycle, and much better outputs were obtained in the 200th cycle. The gradient descent graph including the equilibrium state is given in Figure 6. As seen in the graph, the generator loss function reached the maximum value after the 200th cycle which means divergence. The discriminator loss functions decreased to a small value (0.064–0.09) and generator loss increased to a comparatively high value (1.602), this means the generator cannot be further improved and the learning process has ended.

Intermediate outputs were additionally controlled in addition to the gradient descent graph. Mainly because the learning curve may not be stand-alone determinative, especially in



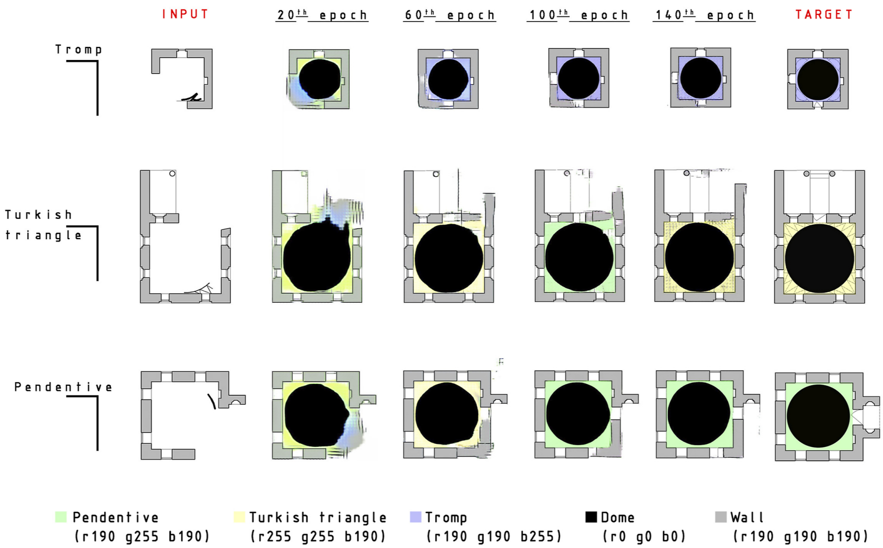
NUMBER OF EPOCHS	1	10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	220
Generator loss	0.7	1.1	0.8	0.6	1.3	0.8	0.9	1.2	0.8	0.8	0.5	0.4	0.5	0.9	1	0.8	0.6	0.9	0.9	0.6	0.5	1.6
Discriminator loss-real	0.4	0.3	0.6	0.4	1	0.2	0.6	0.9	0.5	0.4	0.4	0.3	0.4	0.6	0.6	0.6	0.5	0.2	0.1	0.1	0.1	0.1
Discriminator loss-fake	0.6	0.4	0.3	0.5	0.6	0.4	0.3	0.3	0.4	0.3	0.7	0.8	0.3	0.2	0.2	0.3	0.5	0.3	0.2	0.8	0.1	0.1

Figure 6.
Gradient descent curve including loss function values (logarithmically scaled)

such a sensitive field as architectural conservation. To interpret the results, it is important to notice that one epoch corresponds to training the neural network with all the datasets for once. Since there may be a critical improvement in any epochs of the learning process, outcomes from every pass should be examined in detail. In this context, image outputs from the selected epochs are given in Figure 7 to present the training process. The selection criterion for epochs was to observe a significant improvement in the output.

In the 20th epoch, the learning model was partly effective in generating dome forms, however, in the plan view, there was a remarkably large missing part on the corner of the buildings. In the 60th epoch, the learning model was more effective in generating dome forms, moreover, the missing part on the corner of the buildings was smaller, now. However, the entrance of the buildings could not be generated, completely. The 100th epoch was also similar to the 60th epoch, but the generated dome geometry was in a much finer circular form. In the 140th epoch, the generation of the dome geometry and the entrance was better, but it still had to be improved. The training process continued after the 200th epoch, however, the best results were obtained in this epoch. Regarding both the identification of the wall-to-dome transition types and the production of walls, windows, entrances and domes, the outcome was quite close to the aimed result.

When all the results were evaluated, it was concluded that they largely supported the gradient descent graph, however, this may not necessarily apply in all cases. Since the equilibrium state was reached, the validation tests were performed on the model recorded in this cycle. The missing/damaged states of different tomb plans are transferred as input to the trained model and the outputs are given in Figure 8. It has been observed that the model was successful in recognizing the dome typology and regenerating the dome. The model also recognized the transition elements from the square plan to the dome, but it was not fully successful in identifying the Turkish triangle, due to the finer details. Since the tromp and pendentive transitions have both simpler expressions in the plan view, the outputs were more successful. On the other hand, the proposed framework is developed for the preliminary stages of restitution, only. For example, regenerating the missing geometrical form at a basic



Note(s): Only some of the epochs representing the learning process are selected

Figure 7. Outputs from the training process

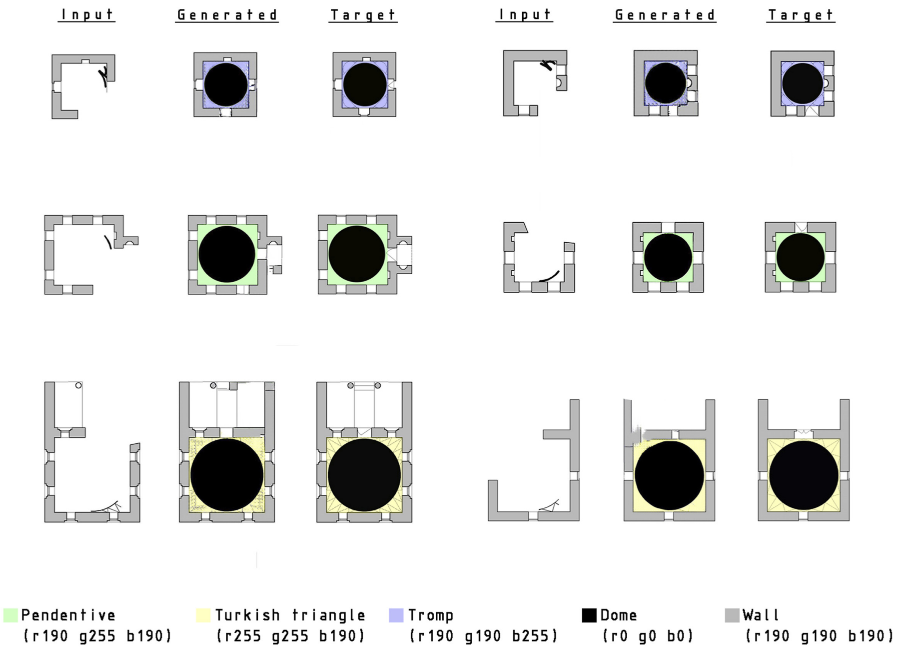


Figure 8.
Validation of the trained model over randomly selected tomb plans

level was an objective. However, for the advanced stages of restitution, the architectural details of the building components including ornaments, materials, identification of construction layers, *etc*. are within the scope. The dataset of this study does not cover these details; therefore, it is not expected to get mentioned architectural details in the outputs.

Following a manual review of the output quality, several evaluation metrics are used to quantify the success of the trained model. These commonly used metrics are "the structural similarity method (SSIM)", "mean-square error (MSE)", and "peak signal to noise ratio (PSNR)". To compare these metrics, several types of degradations ("input image with noise" and "input image with contrast") were applied to the same input image to create comparable versions of it. Then PSNR, MSE and SSIM values were calculated (Figure 9). All of the

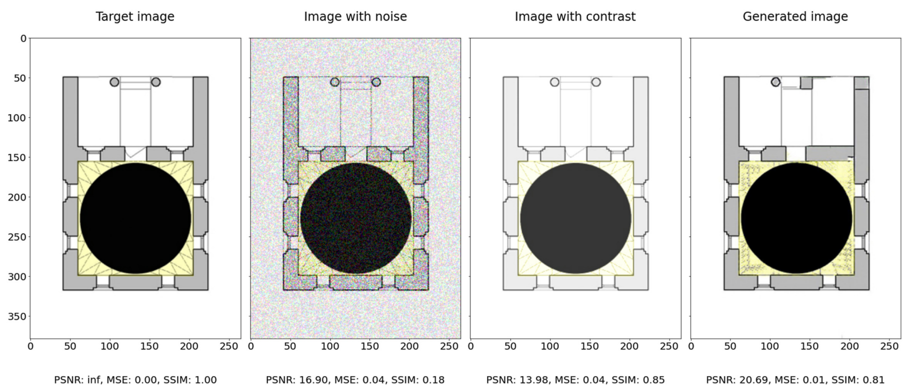


Figure 9.
Comparison of different similarity evaluation metrics over a randomly selected tomb plan

variations returned the same MSE value, showing that MSE could not perform effectively in detecting the difference. Similarly, “input image with noise” yielded a higher PSNR value than “input image with contrast” (a remarkably similar version of the input image) which means PSNR could not perform well, too. The most efficient one was the SSIM in discriminating structural content in the images (the SSIM value of the generated image is 0.81; an SSIM of “0” means that there is no similarity between the images; while an SSIM of “1” means that the images are completely identical).

Simply put, the structural similarity method (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 was chosen because it incorporates loss of correlation, luminance distortion and contrast distortion as a combination of three elements to model image distortion. Accordingly, the trained model was validated with the SSIM over several different tomb plans as the SSIM metric is seemingly indicative to assess the success of the trained model (Figure 10). The yielded SSIM values (0.88–0.95) showed that the trained model is performing well in the preliminary stage of the restitution process of the early Ottoman tombs.

4. Conclusions

The use of computational design and numerical analysis tools in architecture is constantly expanding. The computational design, which started with the parameterization of the design process gained momentum with the development of new techniques and now it provides new opportunities to architects. On the other hand, parallel to the development of these tools, ML

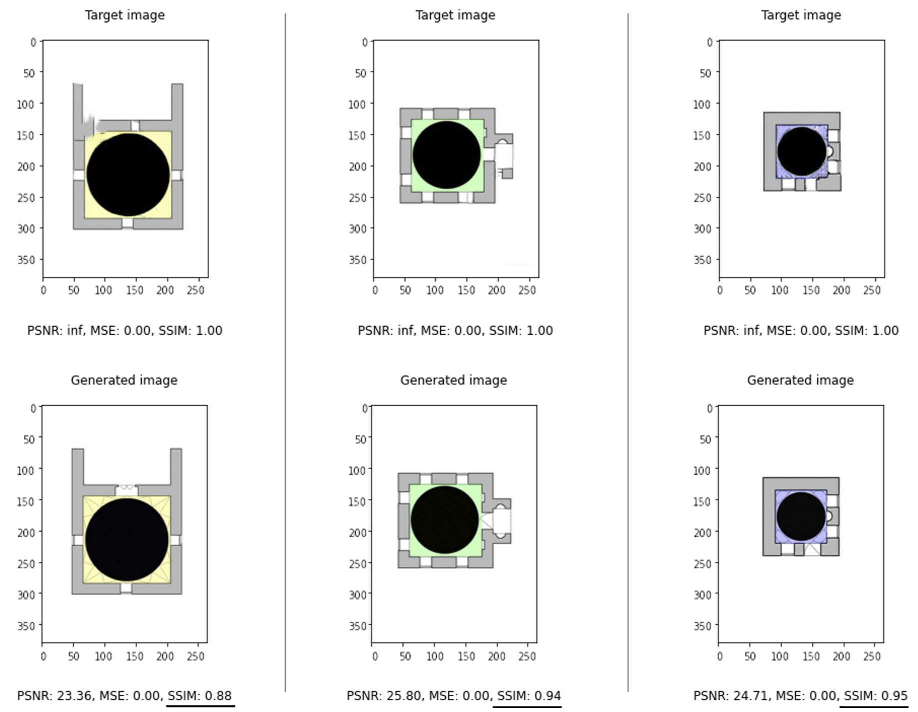


Figure 10. Validation of trained model over several different tomb plans using evaluation metrics

has also started to be of use in the field of architecture and has reached an important widespread effect. Numerous studies have been conducted on the use of ML in the early design phase, and both the potential of this method and the limits confronted have been observed. Therefore, architectural design has started to evolve into a different dimension, especially with the use of large architectural data sets in ML.

The idea that ML can be used effectively in a different field than architectural design has been the starting point of this study. In this context, it is thought that ML can support architects in the preliminary stages of restitution and contribute to the opening of new perspectives in the field. It has been observed that the GAN model used in the study can be beneficial especially in the preliminary stages of restitution when trained on a real data set consisting of plan drawings of many buildings from the same period. At this point, emphasizing the challenges and opportunities and limitations of the proposed method at the case study level is of importance. A reliable outcome has been obtained concerning the interpretation of documented data and the generation of missing data at the macro level. The framework is remarkably effective when it comes to the identification and regeneration of missing architectural components like walls, domes, windows, doors, *etc.* on a macro level without advanced architectural details. On the other hand, the proposed methodological framework was not tested for the advanced steps of restitution since every case of architectural heritage is remarkably detailed and unique. Therefore, the proposed framework for the regeneration of missing components of heritage buildings is limited by the basic geometrical form which means the architectural details of the components including ornaments, materials, identification of construction layers, *etc*. are not covered.

It has been questioned within the scope of the study that the GAN model carries out a pixel-based analysis during the learning phase and sees the data set as image arrays rather than architectural plans, but still obtains reasonable outputs. When the intermediate outputs in the learning process are examined in detail, it is seen that the GAN model can read the repetitive hidden characteristics of the buildings. However, if a data set prepared from different period buildings is used in the learning process, it is anticipated that the outputs obtained will not be as meaningful as those obtained in this study. For this reason, the data set should be carefully prepared by considering the construction periods of the buildings, otherwise, the results may be misleading.

It is also derived that a quantitative metric is necessary to objectively assess the performance of the trained model. Deciding on the metric requires a deep understanding of different metrics. The comparison of the metrics within the scope of this study showed that SSIM is an effective metric to assess the performance of an image-to-image translation method.

Finally, although ML was assessed on the tombs of the early Ottoman period in this study, the proposed flow chart and the mixed methodological framework can be applied to many building typologies, as can be inferred. Conducting studies on different building types using the proposed method will contribute to the verification of the effectiveness of ML in the field. On the other hand, transforming the image outputs into vector drawings will make the data more usable and enable the proposed technique to reach widespread use.

References

Atamturktur, S. and Sevim, B. (2012), "Seismic performance assessment of masonry tile domes through nonlinear finite-element analysis", *Journal of Performance of Constructed Facilities*, Vol. 26 No. 4, pp. 410-423, doi: 10.1061/(ASCE)CF.1943-5509.0000243.

Bassier, M., Vergauwen, M. and Van Genechten, B. (2017), "Automated classification of heritage buildings for as-built bim using machine learning techniques", *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 4 No. 2, pp. 25-30, doi: 10.5194/isprs-annals-IV-2-W2-25-2017.

Bollinger, K., Grohmann, M. and Tessmann, O. (2010), "Structured becoming: evolutionary processes in design engineering", *Architectural Design*, Vol. 80 No. 4, pp. 34-39, doi: 10.1002/ad.1103.

Chaillou, S. (2020), "ArchiGAN: artificial intelligence x architecture", in Philip, F.Y., Mike, X., Neil, L., Jiawei, Y. and Xiang, Wang. (Eds), *Architectural Intelligence*, Springer, Singapore, pp. 117-127, doi: 10.1007/978-981-15-6568-7_8.

Chen, X., Yan, D., Rein, H., John, S., Ilya, S. and Pieter, A. (2016), "InfoGAN: interpretable representation learning by information maximizing generative adversarial nets" in Lee, D., Sugiyama, M., Luxburg, U., Guyon, I. and Garnett, R. (Eds), *Advances in Neural Information Processing Systems 29 (NIPS 2016), Barcelona, Spain*, arXiv:1606.03657.

Chouchoulas, O. and Day, A. (2007), "Design exploration using a shape grammar with a genetic algorithm", *Open House International*, Vol. 32 No. 2, pp. 26-35, doi: 10.1108/OHI-02-2007-B0004.

Coyne, Richard D., Michael, A.R. and Antony, D.R. (1990), *Knowledge-based Design Systems*, Addison-Wesley, Reading, MA.

Croce, V., Bevilacqua, M.G., Caroti, G. and Piemonte, A. (2021), "Connecting geometry and semantics via artificial intelligence: from 3d classification of heritage data to H-bim representations", *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. XLIII B2, pp. 145-152, doi: 10.5194/isprs-archives-XLIII-B2-2021-145-2021.

Duarte, J.P. (2005), "Towards the mass customization of housing: the grammar of siza's houses at malagueira", *Environment and Planning B: Planning and Design*, Vol. 32 No. 3, pp. 347-380, doi: 10.1068/b31124.

Erem, Ö. and Ermiyagil, S.A. (2016), "Adapted design language for anatolian vernacular housing", *Open House International*, Vol. 41 No. 1, pp. 51-58, doi: 10.1108/OHI-01-2016-B0007.

Goodfellow, I.J., Jean, P.A., Mehdi, M., Bing, X., David, W.F., Sherjil, O., Aaron, C. and Yoshua, B. (2014), "Generative adversarial nets", *27th International Conference on Neural Information Processing Systems - (NIPS), Montreal, Canada*, pp. 2672-2680.

Grilli, E. and Remondino, F. (2020), "Machine learning generalisation across different 3D architectural heritage", *ISPRS International Journal of Geo-Information*, Vol. 9 No. 6, p. 379, doi: 10.3390/ijgi9060379.

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, doi: 10.1016/0142-694x(95)00006-d.

Gulli, A., Kapoor, A. and Pal, S. (2019), *Deep Learning with TensorFlow 2 and Keras: Regression, ConvNets, GANs, RNNs, NLP, and More with TensorFlow 2 and the Keras API*, Packt Publishing, Birmingham.

Herr, C.M. and Thomas, K. (2007), "Adapting cellular automata to support the architectural design process", *Automation in Construction*, Vol. 16 No. 1, pp. 61-69, doi: 10.1016/j.autcon.2005.10.005.

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., Jun, Y.Z., Tinghui, Z. and Alexei, A.E. (2017), "Image-to-Image translation with conditional adversarial networks", *30th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2017), Honolulu, HI*, pp. 1125-1134, doi: 10.1109/CVPR.2017.632.

Karaesmen, E. (1993), "Structural behavior of historic masonry domes of major importance: an overview", *Transactions on the Built Environment*, Vol. 4 No. 1, pp. 157-164.

Kuban, D. (1987), "The style of Sinan's domed structures", *Muqarnas*, Vol. 4 No. 1, pp. 72-97, doi: 10.2307/1523097.

Liu, Y., Luo, Y., Deng, Q. and Zhou, X. (2021), "Exploration of campus layout based on generative adversarial network", in Yuan, P.F., Yao, J., Yan, C., Wang, X. and Leach, N. (Eds), *Proceedings of the*

2020 DigitalFUTURES. CDRF 2020. Springer, Singapore, pp. 169-178, doi: 10.1007/978-981-33-4400-6_16.

Llamas, J., M., Lerones, P., Medina, R., Zalama, E. and Gómez-García-Bermejo, J. (2017), "Classification of architectural heritage images using deep learning techniques", *Applied Sciences*, Vol. 7 No. 10, p. 992, doi: 10.3390/app7100992.

Loverdos, D. and Sarhosis, V. (2022), "Automatic image-based brick segmentation and crack detection of masonry walls using machine learning", *Automation in Construction*, Vol. 140 No. 1, 104389, doi: 10.1016/j.autcon.2019.102846.

Mirza, M. and Osindero, S. (2014), "Conditional generative adversarial nets", available at: https://arxiv.org/abs/2012.09457 (accessed 13 January 2021).

Mishra, M. (2021), "Machine learning techniques for structural health monitoring of heritage buildings: a state-of-the-art review and case studies", *Journal of Cultural Heritage*, Vol. 47 No. 1, pp. 227-245, doi: 10.1016/j.culher.2020.09.005.

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*, Vol. 12346, Springer, Cham, doi: 10.1007/978-3-030-58452-8_10.

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, doi: 10.1080/24751448.2019.1640536.

Obeso, A., Benois-Pineau, J., Acosta, A. and Vázquez, M. (2016), "Architectural style classification of Mexican historical buildings using deep convolutional neural networks and sparse features", *Journal of Electronic Imaging*, Vol. 26 No. 1, 011016, doi: 10.1117/1.JEI.26.1.011016.

Ozturk, S., Bayraktar, A., Hokelekli, E. and Ashour, A. (2019), "Nonlinear structural performance of a historical brick masonry inverted dome", *International Journal of Architectural Heritage*, Vol. 14 No. 8, pp. 1161-1179, doi: 10.1080/15583058.2019.1592265.

Pepe, M., Alfio, V., Costantino, D. and Scaringi, D. (2022), "Data for 3D reconstruction and point cloud classification using machine learning in cultural heritage environment", *Data in Brief*, Vol. 42 No. 1, p. 108.

Radford, A., Luke, M. and Soumith, C. (2015), "Unsupervised representation learning with deep convolutional generative adversarial networks", in Yoshua, B. and Yann, L. (Eds), *4th 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", in Gerhard, N.S. (Eds), *CAAD Futures 91 Computer Aided Architectural Design - Education, Research, Application, Zurich, Switzerland*, pp. 285-300.

Turan, M. (1993), "Domes and supports: structural vulnerability of masonry shells", *Transactions on the Built Environment*, Vol. 4 No. 1, pp. 339-346.

Turrin, M., Buelow, P.V. and Stouffs, R. (2011), "Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms", *Advanced Engineering Informatics*, Vol. 25 No. 4, pp. 656-675, doi: 10.1016/j.aei.2011.07.009.

Valero, E., Forster, A., Bosché, F., Hyslop, E., Wilson, L. and Turmel, A. (2019), "Automated defect detection and classification in ashlar masonry walls using machine learning", *Automation in Construction*, Vol. 106 No. 1, 102846.

Wang, N., Zhao, X., Wang, L. and Zou, Z. (2019), "Novel system for rapid investigation and damage detection in cultural heritage conservation based on deep learning", *Journal of Infrastructure Systems*, Vol. 25 No. 3, 04019020, doi: 10.1061/(ASCE)IS.1943-555X.0000499.

Zhang, H. (2019), "3D model generation on architectural plan and section training through machine learning", *Technologies*, Vol. 7 No. 4, pp. 82-96, doi: 10.3390/technologies7040082.

Zhu, J.-Y., Krähenbühl, P., Shechtman, E. and Efros, A.A. (2016), “Generative visual manipulation on the natural image manifold”, *Lecture Notes in Computer Science*, pp. 597-613, doi: 10.1007/978-3-319-46454-1_36.

Corresponding author
İlker Karadag can be contacted at: ilker.karadag@cbu.edu.tr