Systems and Soft Computing 5 (2023) 200051

Contents lists available at ScienceDirect

Systems and Soft Computing

journal homepage: www.journals.elsevier.com/soft-computing-letters

Synthesising 3D solid models of natural heterogeneous materials from single sample image, using encoding deep convolutional generative adversarial networks

Seda Zirek

Bartlett School of Architecture, UCL, UK

ARTICLE INFO

Keywords:
Material synthesis
DCGAN
Texture synthesis
3D Solid textures
Natural heterogeneous materials

ABSTRACT

Three-dimensional solid computational representations of natural heterogeneous materials are challenging to generate due to their high degree of randomness and varying scales of patterns, such as veins and cracks, in different sizes and directions. In this regard, this paper introduces a new architecture to synthesise 3D solid material models by using encoding deep convolutional generative adversarial networks (EDCGANs). DCGANs have been useful in generative tasks in relation to image processing by successfully recreating similar results based on adequate training. While concentrating on natural heterogeneous materials, this paper uses an encoding and a decoding DCGAN combined in a similar way to auto-encoders to convert a given image into marble, based on patches. Additionally, the method creates an input dataset from a single 2D high-resolution exemplar. Further, it translates of 2D data, used as a seed, into 3D data to create material blocks. While the results on the Z-axis do not have size restrictions, the X- and Y-axis are constrained by the given image. Using the method, the paper explores possible ways to present 3D solid textures. The modelling potentials of the developed approach as a design tool is explored to synthesise a 3D solid texture of leaf-like material from an exemplar of a leaf image.

1. Introduction

Natural heterogeneous materials are complex and challenging to regenerate as computational models [1]. Their composition includes several types of elements with different densities and distribution, which are mainly related to the events that occurred during their emergence, making every sample unique [2]. Based on the texture classifications prepared by Pietroni et al., marble is a stochastic texture with a high degree of randomness [3]. Further, a brick wall is a structured regular texture and a stone wall is a structured irregular texture. It is challenging to model solid stochastic textures unless they are scanned and recorded.

Generating 3D solid textures are time-consuming and costly. Additionally, current 2D and 3D material synthesis approaches are generally restricted with an output size. As a result, computer-generated image renderings of large spaces consist of repetitions of small 2D material patches [4]. Popular rendering engines including V-ray and their existing material palettes are in a 2D format. When working with an object expected to have a solid texture with patterns continuing on every face – for instance, when rendering a marble block – if the patterns do not continue from one face to the other because of a lack of 3D solid texture data, visually inconsistent results are created. The method of this paper focuses on generating a cube, from which a desired shape can be extracted to be used for modelling or rendering purposes, which is a common practice in solid texture synthesis. Also mentioned as a type of boundary-independent method, the intended geometry is placed in a synthesised cube of material to obtain the material information specific to each pixel on its surface [3].

The proposed system can be used to generate material blocks consistent with the original input for modelling and design purposes. In addition to the texture generation for computer generated architectural images, there are two practical uses of microscale material data in design; to enhance the creativity of the design process and to design the micro structure of materials for the requirements of design. The first approach of incorporating “design enhancing data” in relation to various aspects of design including performance, materials, and function from an earlier stage has been proven to be beneficial [5]. Mitchell describes design as an ill-defined problem where the production of a solution that cannot be separated from the formulation of a problem [6]. The ill-defined nature of design as a constant evolution of problem and solution space relies on iterative creative processes [7]. The microscale material information can enhance design by informing the design in these iterations. The second approach of designing microstructure of

E-mail address: seda.zirek.12@ucl.ac.uk.

https://doi.org/10.1016/j.sasc.2023.200051
Received 19 January 2023; Received in revised form 10 March 2023; Accepted 10 April 2023
Available online 14 April 2023
2772-9419/© 2023 The Author(s). Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Fig. 1. Selected Italian Marquina marble for dataset creation. Source: Wallpapertag.com.

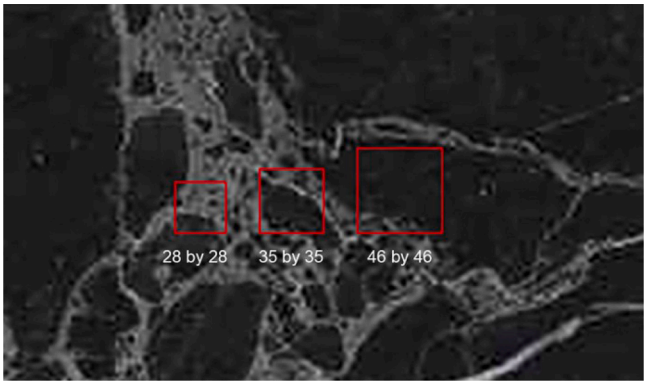

Fig. 2. Square-shaped boundary sizes with edge lengths of 28, 35 and 46 pixels.

materials for the requirements of design is called metamaterial creation [8]. Designing with metamaterials with different behaviours in areas needed is a smart way to use materials from a durability, economy and performance point of view.

There has been extensive research on ML in recent years, particularly regarding generative adversarial networks (GANs) [9]. GANs have been useful in image enhancement and generation. There are many variations of the initial GANs research. This paper focuses on an unsupervised version of GANs using a convolutional neural network (CNN) architecture called DCGAN [10]. DCGAN can generate images from an input of noise. Some of the impractical aspects of the algorithm include the need for a high number of samples to generalise the features and that typically an image or a model of fixed size is created. The method developed in this paper proposes to solve both issues by first introducing a new approach to create an input dataset from a single high-resolution image of marble. It analyses a given exemplar to generate an entire dataset for an ML algorithm to learn to create similar 3D solid textures while eliminating the issues of inconsistencies in solid textures. Secondly, it proposes two separate solutions concerning the X-, Y- and Z-axis to accomplish results independent of size; the X- and Y-axis are constrained by the given image and the Z-axis is generated using the latent representation. The paper proposes to use two DCGANs: one reversed (encoder) and one normal (decoder). They are merged in a similar way to auto-encoders, to create a generative system that synthesises marble photos based on a given layout. While, the encoder and the decoder sections are trained separately, the latent representation of this new architecture, the EDCGAN, is prepared by the decoder following its training. This way, tile-by-tile, the EDCGAN can copy the patterns of a given image to overlay a marble look. To obtain a size-independent solution on the Z-axis, the proposal translates 2D tiles into the 3D space by gradually changing the noise data.

The main objective of the paper is to create a designer's tool that can create size-independent models of natural materials. It utilises this sample-based texture-synthesis method for modelling and design purposes to translate 2D images into 3D models, demonstrated on an exemplar of a naturally flat leaf image to synthesise a block of leaf-like material. The system creates a 3D leaf block using the original 2D leaf image. The veins of a leaf, evenly distributing resources including water, continue vertically and change gradually, creating a partition system on a non-standard and non-uniform grid. To investigate the results further, the paper extracts the geometry of the leaf partition from the rest of the block. The isolated section demonstrates a balanced non-uniform grid geometry that gradually changes on the vertical axis. Although the method presented in this paper is not designed to optimise synthesised material blocks for specific solutions, this could be indirectly achieved by choosing the right seed image.

One of the main challenges of ML work is to obtain a workable dataset. The approach introduced to create an input dataset from a single high-resolution image is beneficial in terms of making it easier to obtain a dataset for any other material using a single high-resolution photo. Another obvious benefit is that it keeps the style of the marble consistent with the original image. The sufficiency of one high-resolution image to create an entire dataset increases the efficiency of the workflow.

2. Literature review

In the field of 2D image and texture generation on a larger scale, a study by Wei and Levoy accomplishes a Markov Random Fields-based method that synthesises realistic textures despite with an output size

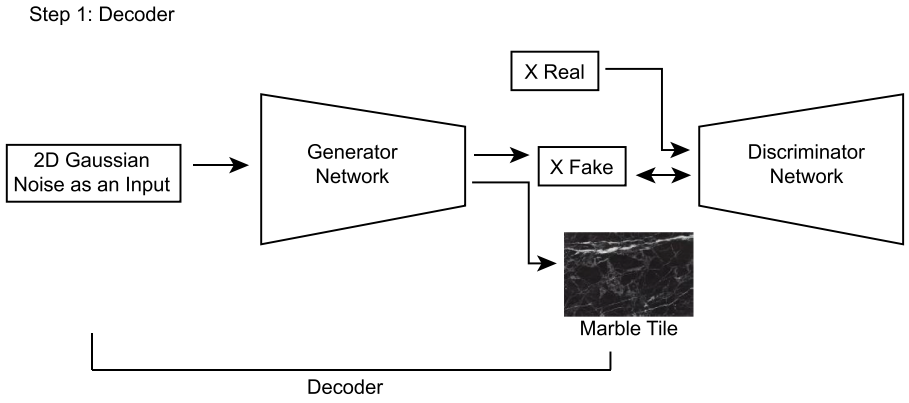

Fig. 3. The architecture of the decoder. The generator part will be used as decoder.

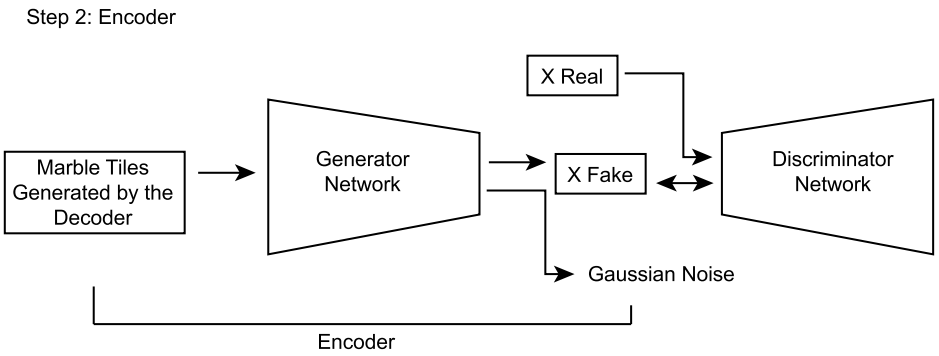

Fig. 4. The architecture of the encoder. The generator part will be used as encoder.

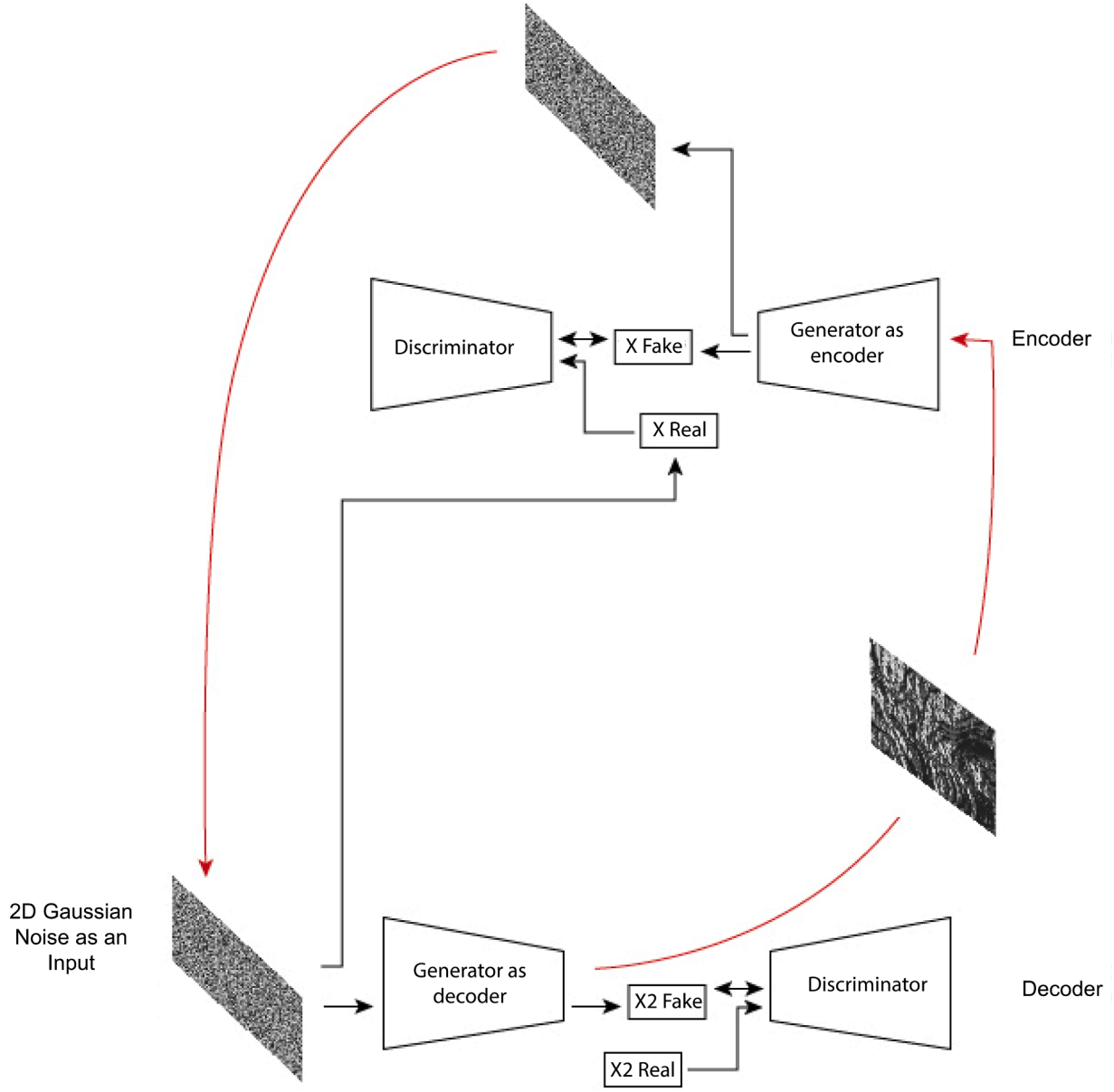

Fig. 5. The architecture of the EDCGAN.

restricted to double the size of the input [11]. Furthermore, Lefebvre and Hoppe use a parallel synthesis method to create infinite aperiodic patterns [12]. Following the advances in ML algorithms, a study by Bergmann et al. uses a technique of blending in combination with GAN to achieve endless image composition [13]. Nevertheless, while the results of these studies are satisfying in terms of generating realistic patterns, the methods cannot be used in synthesising most of the heterogeneous natural materials. A white vein in Italian Nero Marquina marble

Table 1
Architecture of the generator section of the DCGAN.

dense_12 (Dense)	(None, 6272)	4923520
reshape_8 (Reshape)	(None, 7, 7, 128)	0
up_sampling2d_15 (UpSampling	(None, 14, 14, 128)	0
conv2d_36 (Conv2D)	(None, 14, 14, 128)	147584
up_sampling2d_16 (UpSampling	(None, 28, 28, 128)	0
conv2d_37 (Conv2D)	(None, 28, 28, 64)	73792
conv2d_38 (Conv2D)	(None, 28, 28, 1)	577

continues until it gradually dissolves and timber has annual rings gradually changing their thickness to make space for emerging knots. These characteristics cannot be reproduced using aperiodic approach based on repetition.

Material data which includes the interior configurations along with the faces is called a 3D solid texture. One of the early studies on solid textures by Perlin proves the effectiveness of such materials in realistic renders [14]. Perlin's study is limited to isotropic materials due to a function-based approach to synthesising its patterns. Kopf et al. classify methods to synthesise solid textures under three categories: parametric, non-parametric, and optimisation-based [15]. Parametric approaches require a set of rules to create the texture. Non-parametric approaches solve the overall configuration by calculating each pixel individually. Optimisation-based approaches generate the final pattern through iterations of development. Kopf et al. utilise a global histogram-matching method with a non-parametric algorithm while successfully creating solids from 2D images [15]. ML-based material synthesis can be considered as the fourth category, where a model is trained to generate sections of a consistent result.

ML has been used extensively in generative image modelling. Algorithms like GANs, variational inference, and autoregressive are some of the well-known techniques. An example study by Zhang et al. focuses on creating large images using GANs, while omitting a tiling-based approach, by stacking two GANs to first create a low-resolution image and secondly to create a super-resolution image [16]. The dataset used for this research is MS COCO, containing 80 K samples. Although the results are encouraging, it cannot create an image of infinite size.

An example of generative image modelling in 3D space is by Dosovitskiy et al., who trained a CNN with rendered 3D models of chairs [17]. The results are successful in terms of acknowledging the capacity of these networks for learning and interpolating shapes of the real world in 3D. The resulting network of the study can generate new chairs by combining learned examples from the training set. A study on 3D texture synthesis using GANs by Zhao et al. successfully demonstrates a GAN-based algorithm creating textures similar to the given exemplars [18] while being limited to aperiodic textures and size restrictions. Another influential work by Portenier, Bigdeli, and Goksel utilises Gram matrices of features to synthesise 3D solid textures using GANs [19]. An on-demand approach to texture synthesis by Gutierrez et al. utilises a CNN generator focusing on isotropic and anisotropic textures [20]. These studies generally omit heterogeneous natural materials with varying scales of patterns in different sizes and directions. In particular, the pattern distribution in marble is random, with some areas almost blank and others with high concentrations of patterns. The continuity of the features, as in the veins of marble, of the synthesised materials of the studies has not been fully explored in 3D space.

Transfer learning is an ML method that utilises knowledge gained from a previously used model to extend training to perform a similar task. The study by Bostanabad focuses on transfer learning to synthesise 3D models based on 2D images of irregular patterns [21]. It proposes a system that is quick to train with minimal computational cost, and a pre-trained CNN transfers the 2D data into a 3D material, where the size of the synthesised blocks is limited to the size of the neural network. Increasing the size of the output is computationally costly as this requires the network architecture to be enlarged. Another closely related study focuses on synthesising non-stationary textures with large-scale,

Fig. 6. The results created by the decoder trained with 28 × 28 pixel tiles obtained from the original marble image above. The decoder successfully synthesises marble tiles.

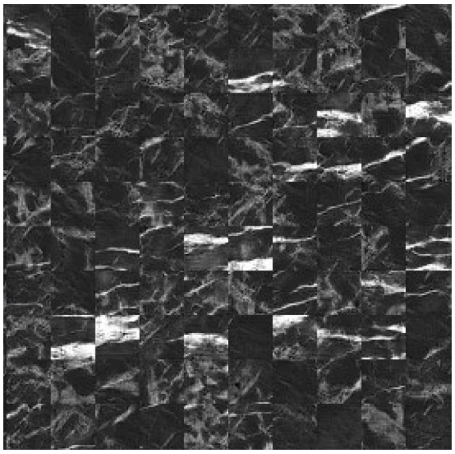

Fig. 7. Random tiling trial, using images generated by DCGAN.

spatially variant, inhomogeneous structures, which can only double the size of an input image while being faithful to the original features [22]. Other recent studies include Chen et al.'s work on stationary patterns using an implicit periodic field network [23] and Henzler, Mitra, and Ritschel's work on creating spaces of textures addressing the issues of variety [24]. The latter study concentrates on the noise of VGG activations with the Gram matrix to create variety in results. However, like the previously mentioned studies, the continuity of the patterns, as well as the elimination of size restrictions, has not been explored.

To summarise, the previous studies on solid texture synthesis generally aim to work on homogenously porous materials with small aperiodic repetitive patterns. In contrast, natural heterogeneous materials, with varying scales of patterns in different sizes and directions, are difficult to model. Further, the studies that do focus on natural heterogeneous materials are characterised by size restrictions, originating from either the model or the input. Therefore, the synthesis of 3D models of natural heterogeneous materials, including their specific traits, such as veins and cracks, with no size restrictions, has not been fully explored. In this context, the main aim of this paper is to synthesise size-independent natural heterogeneous material blocks containing patterns, such as cracks and veins, in various sizes and directions.

Finally, there has been extensive work in the area of dataset enlargement from limited resources. When the source data is an image, the conventional methods of pixel interpolation, including pixel replication and cubic-spline interpolation and ML-based methods of single-frame super-resolution, are amongst the successful studies [25]. Generally referred to as a super-resolution problem, learning-based methods enhance low-resolution images using supervised datasets, which consist of pairs of images with the low-resolution version as an input and the high-resolution version as an output [26]. Furthermore, data augmentation is an iterative method based on computing posterior distributions to enlarge datasets [27]. Recent research shows that GANs can also be used for data augmentation [28]. Two closely related studies of training GANs from a single exemplar utilise the models of SinGAN and InGAN [29,30]. The first study is based on learning patch distributions via a pyramid of GANs, where each GAN is responsible for a specific scale of patch. This scaling approach, also existing in the second study, is useful to learn features in different sizes. Consequently, the dataset prepared for this paper utilises a single high-resolution marble image by visiting every pixel to capture its surrounding in three different scales, which are normalised by downsizing the captured area into a format of unlabelled 28×28pixel tiles.

3. Approach

3.1. GANs

First presented in a paper in 2014 by Goodfellow et al., GANs are based on two networks working against each other: a generator and a discriminator [9]. The first network learns to generate images, and the second learns to detect the mistakes in these images. Therefore, after sufficient training, the results of the generator begin to pass the realness check of the discriminator. Hence, the generator becomes better at creating images of increasing verisimilitude. Based on this structure, a sufficiently trained generator takes noise images as an input and creates images similar to the input dataset as an output. This paper is based on a DCGAN example by Linder-Norén [31].

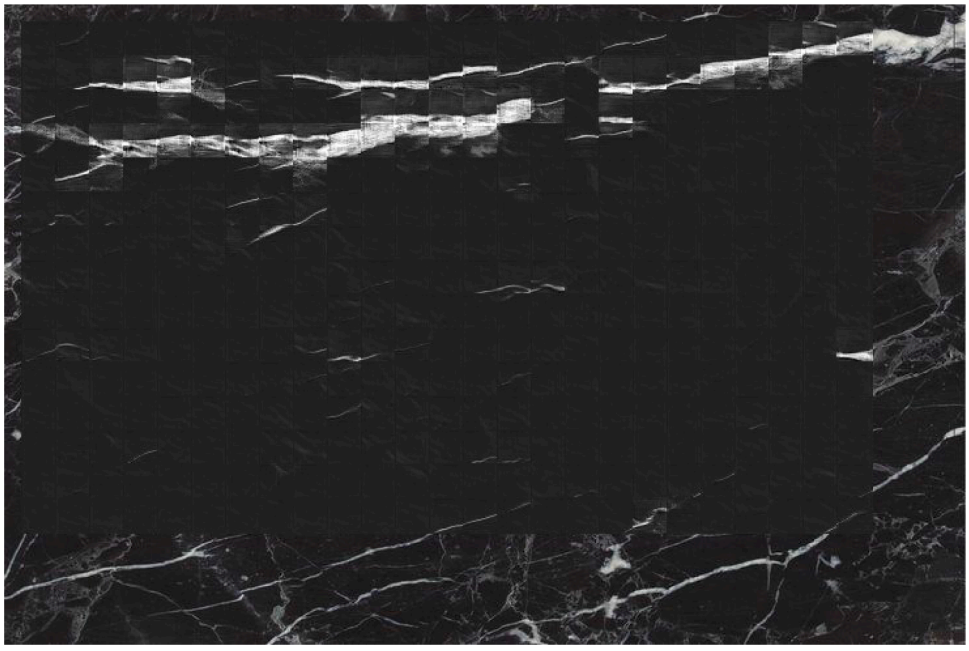

Fig. 8. The new predictions of the EDCGAN are placed back on the original marble image separated by a white frame. The synthesised section repeats the original image, while eliminating some of the less obvious patterns.

Fig. 9. The close-up wood photo used to generate marble.

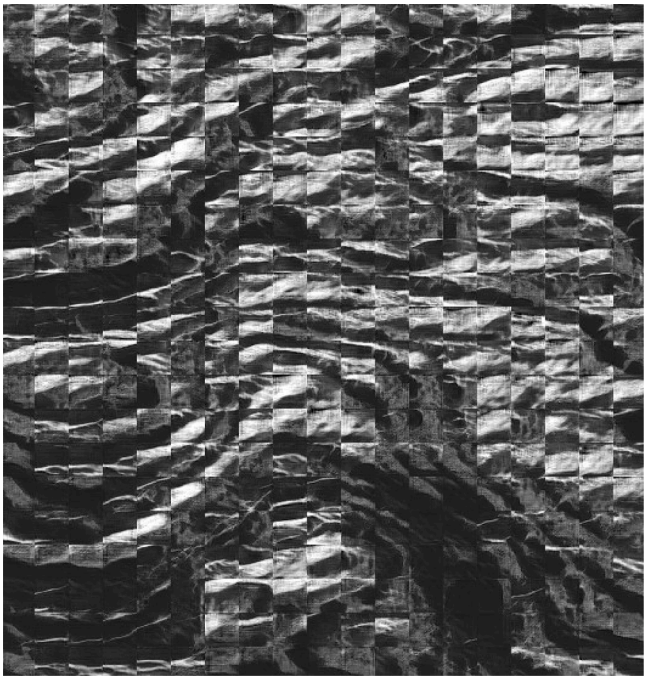

Fig. 10. A marble image generated using the close-up wood photo above. Although each tile resembles marble and the overall configuration of the image follows the grain of the wood, the results are not continuous, and the transition between any two tiles is abrupt.

3.2. Marble

The selected marble type of Nero Marquina from Italy has intense interior ornamentation and its colour range is monochrome as shown in Fig. 1. The studies in this paper are limited to grayscale for simplicity purposes. However, the approach can easily be applied to colour images without any significant modification, yet increasing the computational load.

Marble has isotropic qualities with matching layouts in all directions, due to its random nature. For instance, a slice from the X- and Y-axis is similar to a slice from the Y- and Z-axis. By looking at a single face image

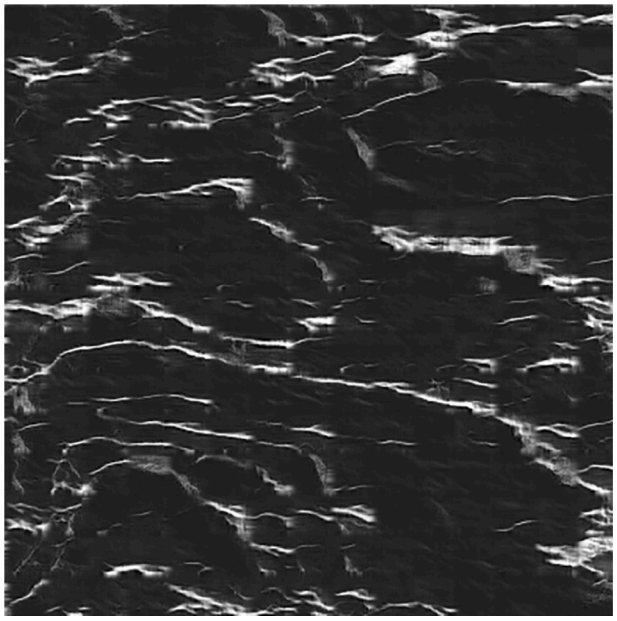

Fig. 11. A marble image generated after the second iteration, aiming to minimise the interruption between the tiles. However, the blending is not fully resolved.

Fig. 12. The second close-up wood photo with intricate patterning used to generate marble.

of a block, the interior structure of the hidden areas can approximately be predicted. This method is commonly used by tile manufacturers for material selection purposes prior to the purchase of blocks of raw materials [32]. Generally, the methods investigated in this paper require isotropic qualities, to effectively switch from a two-dimensional space into three dimensions and to predict or generate material blocks from a single image.

The first step of working with DCGANs is to prepare the dataset consisting of 28×28pixel tiles, each one containing a section of the high-resolution marble image. The paper follows a pipeline of setting up an ML algorithm involving dataset creation, constructing a model for training and testing, and image generation.

3.3. Dataset creation

This paper confines itself to using a single face image of a marble block for the dataset creation. The preparation process is based on a logic similar to cellular automata (CA), where each cell in a set of cells is

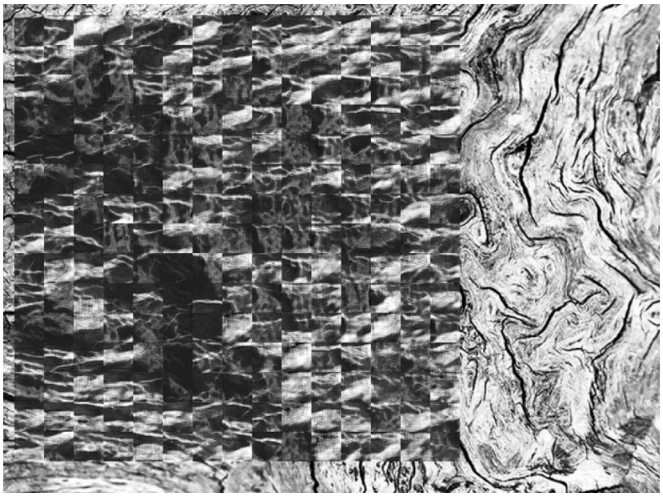

Fig. 13. EDCGAN trained with a marble image modify a section of the timber image. The result is a blend of two images while following the underlying pattern of the grains and a local marble appearance per tile.

relative to its neighbours [33]. CAs are commonly used in material sciences, particularly for the simulation of particles [34]. The formation process of marble, which is a stone created through the metamorphism of carbonate mineral grains, exhibits similar features to CA. Its spatial autocorrelation means the shape, colour, or size of a marble composite are dependant on the shapes, colours, and sizes of its neighbours [35]. Using the analogy of the mineral grains to the cells of CA, each pixel on a marble image is strongly related to its immediate neighbours.

Marble contains patterns in different sizes and directions. To teach an algorithm to generate a particular material, the input data should reflect the character of that material consistently. While using a CA-like logic to create a dataset from a single marble image, "for each cell, a set of cells called its neighbourhood" should be defined and this neighbourhood should reflect the character of that material. Selecting the right size of neighbourhood boundary can be challenging, as the size and shape of the patterns are different in each marble. For instance, if we are looking at too small an area, we might miss some of the patterns which are essential to that type of marble – namely, distinct patterns to identify its type. To choose the right neighbourhood boundary, which includes most of the distinctive patterns of a particular marble image, this paper uses a system with three different square-shaped boundary sizes with edge lengths of 28, 35, and 40 pixels as shown in Fig. 2. These different scales are downsized to 28 × 28 pixels for standardisation. Finally, using the original marble image of 736 × 490 pixels, which is then multiplied with the three different scales, resulting in a dataset of 1081,920 tiles is created.

3.4. Constructing the model for training and testing

To achieve size independence, the paper proposes combining two DCGANs – a decoder and an encoder – into a single system called EDCGAN. The proposed model is inspired by variational auto-encoders (VAE) and VAEGAN [36,37]. The two DCGANs are trained separately with the decoder preparing the dataset of the encoder. Once the training is complete, the combined system can turn tiles obtained from an image into marble.

3.4.1. Decoder

Conventional auto-encoders automatically generate the latent representation. Here, the paper initialises a set of 2D Gaussian 28 × 28 noise samples and uses the set as an input to the decoder DCGAN. This is the latent representation provided to the decoding step. The training set consists of 865,536 28 × 28 Gaussian noise tiles as input and 28 × 28 marble tiles as output, and the testing set comprises 216,384 pairs of noise/marble tiles. The target of the decoder is 28 × 28 marble tiles. Fig. 3 demonstrates the architecture of the decoder. In both decoder and encoder, X Real refers to the output of its training dataset. X Fake refers to the generated tiles during training.

3.4.2. Encoder

The paper trains the encoder by using the images generated by the trained decoder, targeting the initial Gaussian noise set. The training set consists of one million 28 × 28 generated marble tiles as input and produces 28 × 28 2D Gaussian noise tiles as output, and the testing set comprises 200,000 pairs of marble/noise tiles. The target of the encoder is 28 × 28 2D Gaussian noise tiles. Fig. 4 demonstrates the architecture of the encoder.

3.4.3. EDCGAN

When two networks are put together with the encoder first and the decoder second, creating an EDCGAN, noise is generated from a given tile as the latent representation, and this noise is then changed back into a marble tile. The approach differs from the conventional auto-encoder approaches that create latent representation automatically as part of the training process. Fig. 5 demonstrates the architecture of the combined system.

In summary, the decoder is trained with the initial dataset, while each time the generator section of the network creates an image, the discriminator rates the result based on its marble-likeness. This rating is then transferred back to the generator to do a better job. Eventually, this process leads to a generative network that knows how to create a believable marble image from 2D Gaussian noise tiles. Once the training is complete, to create the dataset of the encoder, the generated noise and marble tiles are recorded. The encoder is trained with the new dataset to turn marble tiles into noise. After training, both DCGANs are combined in a reversed order, with second as first and first as second. The consequence is a combined system that can turn a tile into noise, and from that noise a marble tile is generated. If the source image includes continuous features, despite being unrelated to the patterns of marble, the system can generate patterns following the layout of a given image to create marble-like texture with some segmentation between the tiles. Generally, a trained DCGAN can only hallucinate what it is trained with – the only thing it knows. Hence, if a model is trained with marble images, after training, no matter what the input, it will only produce marble patterns. An example of such an approach was studied using deep neural networks (DNNs) and applied to different objects to create dream-like images of animals, vehicles, etc. [38].

Table 1 shows the architecture of the generators, which are a type of CNN. Both the encoder and the decoder share the same architecture. Several trials aiming to reduce the size of the models to decrease the training time resulted in a significant loss in learning. Therefore, the demonstrated architecture has been selected as the optimum size of the networks.

4. Investigating and visualising

The results of the decoder, creating marbles from Gaussian noise, show us that it successfully synthesises images of marble. The size of the end result is the same as the training set images of 28 × 28 pixel tiles. Fig. 6 shows the original marble image above and the generated tiles below. The steps of this process involve training a DCGAN, extracting the generative section, creating an input dataset consisting of 2D Gaussian noise tiles, and feeding each input to the extracted generator to obtain the new marble tiles.

Fig. 7 demonstrates alarger image with randomly configured and composed tiles obtained from the decoder to observe the generative skills of the DCGAN. Although the tiles are not correlated and create segmentation, the overall image resembles an appearance of marble.

The second step is to create a dataset using the decoder, to train the

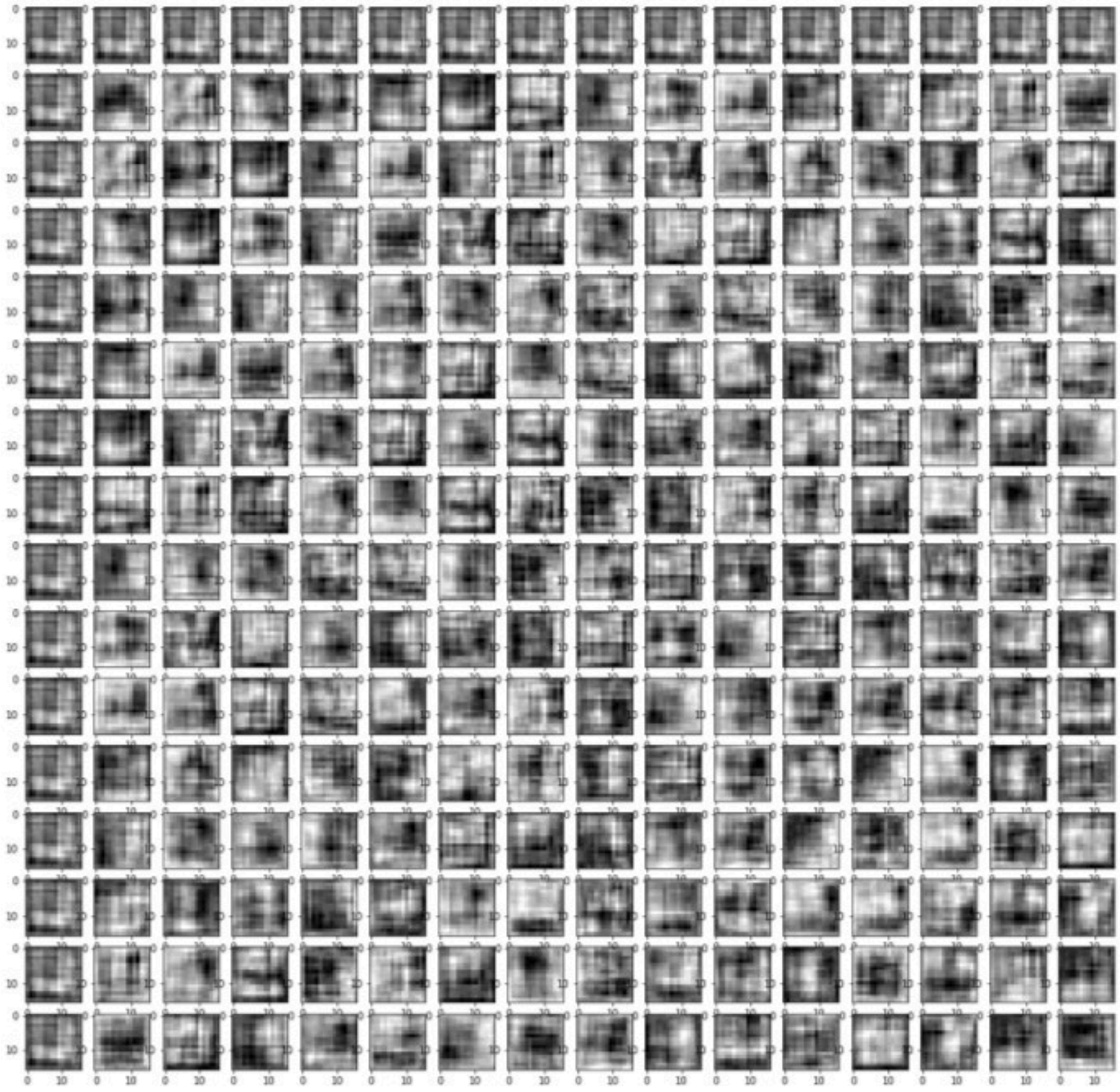

Fig. 14. The features extracted from the decoder.

encoder in reverse function. Once it is trained, the encoder creates the targeted noise of the associated marble tiles. The combined system first changes the images into noise and the noise back into marble tiles. Using this system, the paper can synthesise results constrained by the size of the base image. Some segmentation due to the tile based generation can still be observed. To test, while using the original marble image as a base, the synthesised tiles are placed back in the same location. Fig. 8 with 736 × 490 pixels demonstrates that this setup can repeat the original image, while eliminating some of the less obvious patterns and only recognising the major ones. The seams between the tiles are visible but the major white vein continues from left to the right of the image.

4.1. Trained EDCGAN applied on other image layouts

As a second trial, the trained system is applied to a wood image containing a close-up view of its grains as continuous features. Similarly to marble, wood has a heterogeneous structure and a limited colour range. Hence, it can conveniently be applied as a base image. Fig. 9 shows the original photo, while Fig. 10 shows the results of the first trial.

The size of the image demonstrated in Fig. 9 is 574 × 615 pixels. Once the EDCGAN is applied to generate Gaussian noise from the given image and marble tiles from the Gaussian noise, the result is 560 × 532 pixels, consisting of 380 28 × 28 tiles. Although each tile resembles marble and the overall configuration of the image follows the grain of the wood, the results are not continuous, and the transition between any two tiles is abrupt. To address this issue, the resulting image is reapplied to the system for a second iteration. In this second iteration, the tiling is slightly shifted to fix the segmentation. Fig. 11 shows the final image with minimised interruption between the tiles and a continuous global patterning following the grain of the wood. Nonetheless, the paper

Fig. 15. An investigation of spatial correlation between noise and its synthesised tile created by the decoder.

Table 2
Sizes of the groups in pixels.

100% of the latent representation is $28 \times 28 = 784$ pixels per tile.
Approximately 10% of the latent representation is 78 pixels per tile.
Approximately 20% of the latent representation is 156 pixels per tile.
Approximately 40% of the latent representation is 313 pixels per tile.

acknowledges that even in the second iteration, the blending amongst tiles is not fully resolved. Additionally, the marble appearance of the final synthesised image is limited due to global patterns originating from the wood grain.

A third trial applies the EDCGANs on a different close-up image of a log of wood with intricate patterning as shown in Fig. 12. Similarly, the generated tiles are placed back in the same location, as demonstrated in Fig. 13. Parts of the original image are untouched for comparison. The synthesised section is a blend of two images while following the underlying pattern of the grains and a local marble appearance per tile.

4.2. Further investigation focusing on the decoder

There have been several studies to understand how CNNs recognise patterns and why they are so powerful. Zeiler and Fergus focus on regenerating the learned hidden patterns by attaching a deconvolutional network to every layer of the original CNN. They can observe which areas are activated and what patterns are created [39]. To visualise the learned features of the marble synthesiser, a deconvolutional network is attached to the last layer of the decoder to produce the results seen in Fig. 14.

Furthermore, an exercise to see a possible correlation between the location of the noise data and the actual image is pursued on the decoder. In detail, a subset of noise data is selected and deformed gradually by deleting and adding a new random bunch to the end of each data individually. Fig. 15 shows the result of the trial with 10%, 20%, and 40% change to the original noise, demonstrating a lack of spatial correlation between the noise and the resulting tile. Table 2 demonstrates the pixel equivalence of the percentages. Specifically, changing the noise data on a particular side of a tile does not result in changes on that particular side of the synthesised marble. In the following trials, the noise is changed from the right side of the tiles and holistic changes in the results are observed. It is also noteworthy that 10% change is not very significant, but that 40% change creates completely different marble tiles.

4.3. Creating a three-dimensional marble block

This section focuses on enhancing the results of the EDCGAN from 2D into 3D space. Using the two trained networks, to change the base image into noises and the noises back into marble, a sequence of tiles enabling the transition to the third dimension is generated. Figs. 16 and 17 demonstrate a gradual change, which is the main characteristic of the sequences, that are accomplished by gradual change in their related noise data.

Once the sequence is created, these images are repositioned in 3D space to create a material block. In detail, the patterns of the tiles refer to the different elements with various densities [40]. The 3D model should accurately present the complex composition of the solid texture data. Hence, a series of studies aiming to find the most suitable modelling method to present the data have been investigated. The following presentation studies are introduced in an order starting from simple to complex.

In Fig. 18, the pixels of the tiles are separated into two sections using a threshold greyscale value of 90. Pixels below the threshold are darker and omitted in the model. Pixels above the value are lighter and present the veins of the marble tiles. Hence, they are incorporated in the model as a point cloud. The boundaries of the marble cube are marked as a 3D

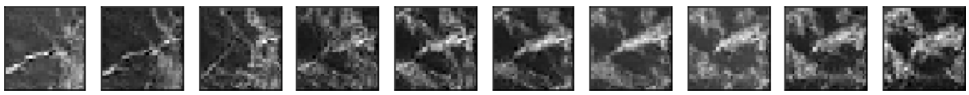

Fig. 16. An example sequence of images to generate a marble block.

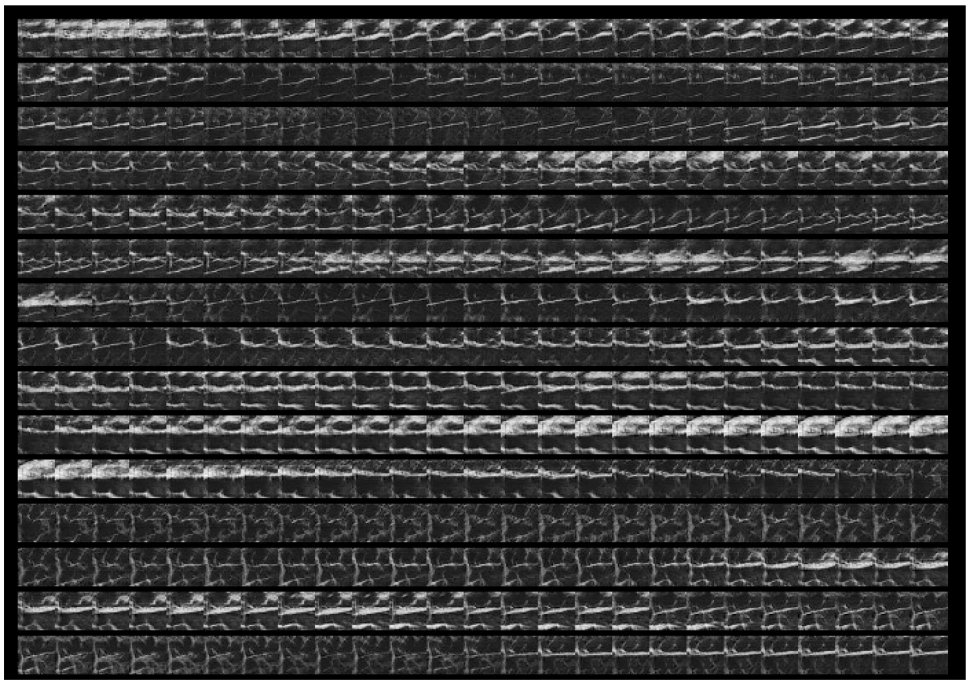

Fig. 17. An example sequence with 375 marble tiles.

grid.

The second trial focuses on increasing the diversity of different areas with various greyscale values. Similarly to the first trial, a grouping method based on thresholds is used. Five groups to differentiate the material and to generate the marble cube are considered to be sufficient without overwhelming the readability of the results. Furthermore, a basic point-cloud presentation with colouring of groups is inadequate. Hence, 3D geometries per each isolated group are created using the N-particle to polygon conversion function of Maya software. Figs. 19 and 20 show the groups of materials composed in the final block. Fig. 21 demonstrates the original sequence of tiles.

In this implementation, the upper limit to grouping based on thresholding is 255 as each pixel's colour value can only range between 0 and 255. However, this study aims at finding a suitable presentation method to maximise readability. The methods selected should show the complexity of the material with intertwined clusters.

Fig. 22 presents the data using a pixel-based approach, where each point in the cloud is coloured according to its greyscale value. Therefore, this approach is an example of the highest grouping number of 255. However, the groups cannot be converted into geometries as they are too dispersed. Moreover, the resulting marble block partially show the complexity of its interiors. Traits such as veins can be observed as continuous lines running across the 3D solid texture. The second presentation method with five groups is selected as the most suitable approach to the task of marble cube generation.

4.4. Variable density material block

Following the example applications of EDCGAN, this section uses the same system to generate a non-existing material from a close-up image of a leaf as shown in Fig. 23 Once the models are trained with tiles obtained from an exemplar, they learn to repeat the underlying logic implicit in the original photo. Therefore, by teaching the models to make a leaf, they learn how the paths of the veins are distributed to evenly dispense water and other resources throughout the surface, called leaf venation [41]. The underlying functions that shape a leaf are repeated throughout the synthesised material block, layer by layer. This study creates a new material, influenced by the geometrical logic of a 2D leaf image expanded into 3D space. The resulting geometry does not resemble the cross section of a leaf as leaves have anisotropic characters. Hence, this experiment shows that the proposed method is limited to isotropic materials with similar patterns on all directions. Indirectly, the developed algorithm that synthesises 3D material blocks accomplishes two goals:

1 Repeating a structure without needing to know the hidden rules behind it.
2 Transforming and enlarging this structure.

The process to turn a leaf image into a 3D material block consists of creating the dataset of tiles, training the EDCGAN, generating a sequence of leaf images, and creating a 3D material block from the sequence. This approach enables a size-independent material generation on a single axis. As shown in Fig. 24, the sequence of tiles to generate the leaf block can be as many as needed. The resulting block demonstrates a gradual change and an adequate leaf pattern on every level. Hence, the generated material block extends the logic of a leaf vein system into the three-dimensional space.

The system creates a 3D leaf block using the original leaf image. The veins of a leaf, evenly distributing resources including water, continue vertically and change gradually, creating a partition system on a non-standard and non-uniform grid. The resulting model is generated using the pixel-based technique investigated in the previous section. The greyscale leaf images are separated into two groups based on a greyscale

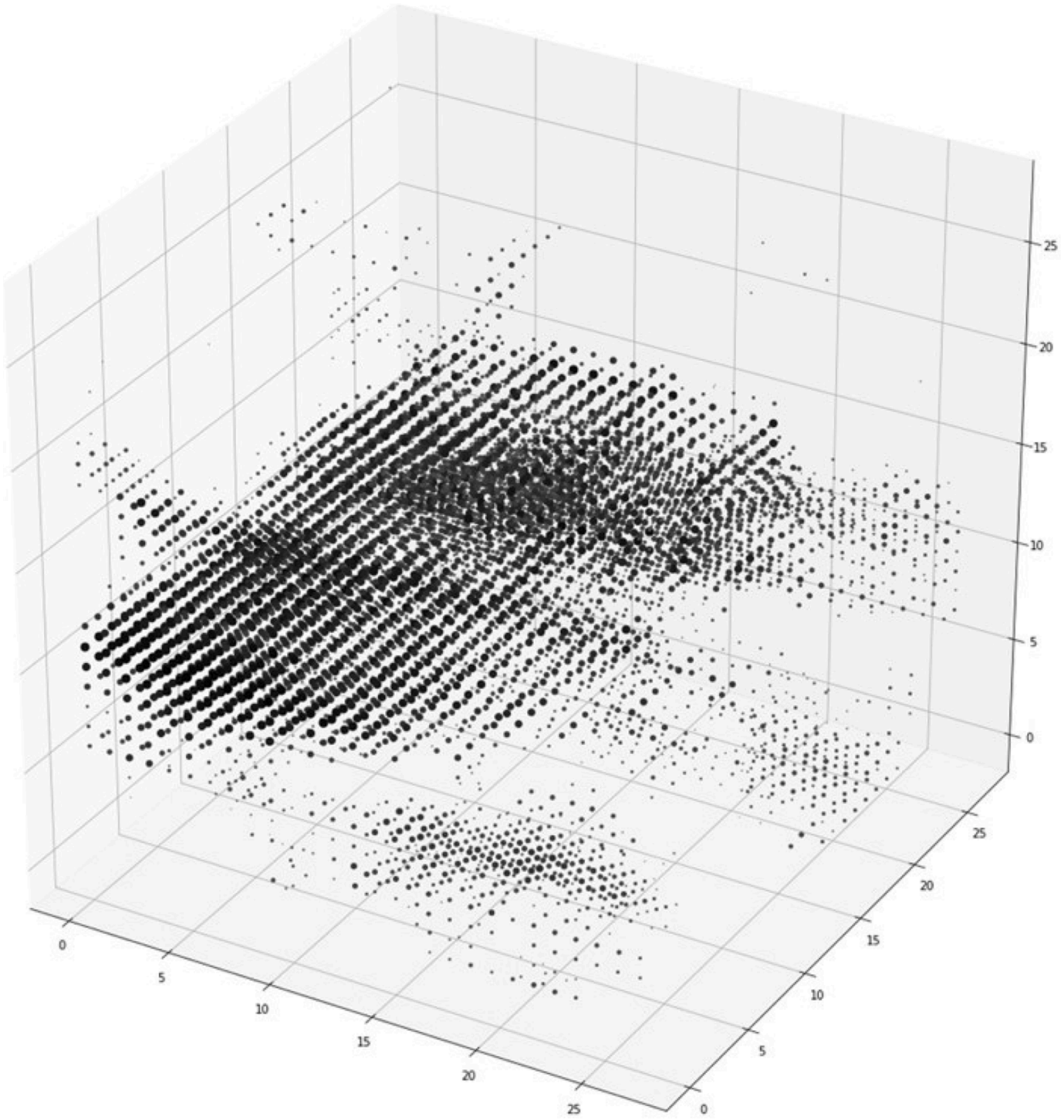

Fig. 18. An initial experiment on visualising the 3D marble block using a point cloud.

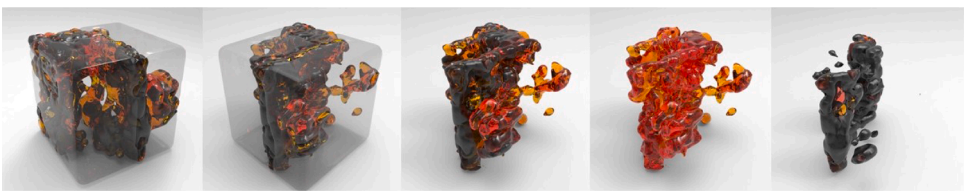

Fig. 19. 3D geometries of groups composing a marble block.

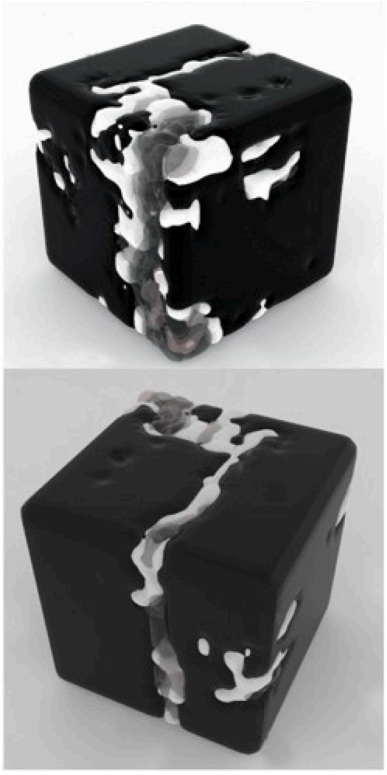

Fig. 20. A 3D model cube generated using five groups with various greyscale values.

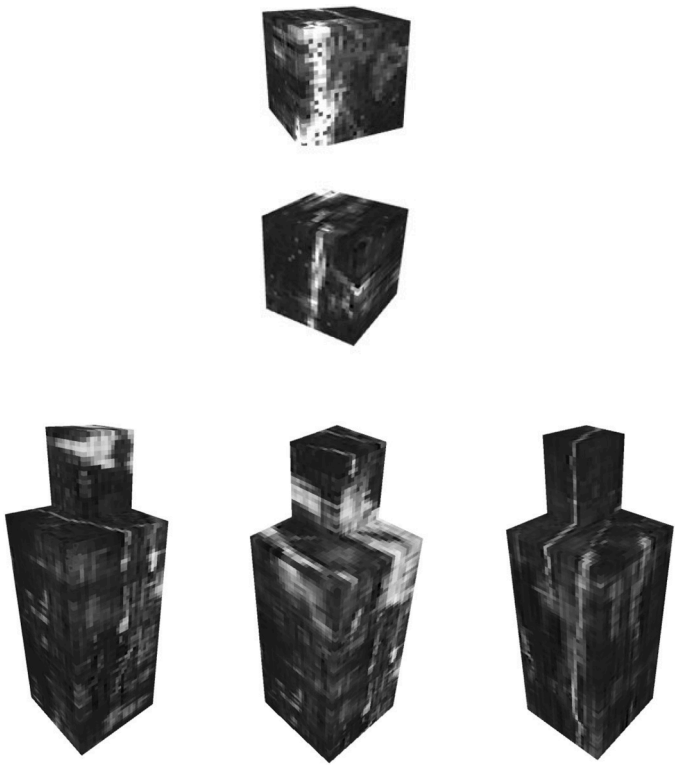

Fig. 22. A 3D model generated using a method of colouring each pixel with its greyscale value. Traits such as veins can be observed as continuous lines running across the 3D solid texture.

threshold value of 90. Lighter areas, which are the representation of the leaf veins, are selected to include in the final model. Fig. 25 shows the rendering of the leaf block.

Figs. 26 and 27 show a3D printed version of this model to test the connectivity and continuity of the geometry, which proves that the final geometry works as a single entity. The pixels above the greyscale value of 90 are connected to the next level at each layer, resulting in an uninterrupted and continuous model. Moreover, the grid originating from the leaf structure is gradually changing on the vertical axis. Finally, the block is stable and stands on its own.

5. Conclusion and future works

This paper presents a method to create 3D textures from a single high-resolution exemplar using EDCGANs. The proposed method utilises two DCGANs, where the decoder is used to prepare the dataset of the encoder. The method can be used to synthesise material images or blocks from a base image with continuous features. The paper focuses on marble, as it is a natural heterogeneous material, which is generally challenging to model due to its random nature. The presented study is limited to isotropic materials. While the results on the Z-axis do not have a size restriction, the X- and Y-axis are constrained by the given image. The paper accomplishes the continuity of features, such as cracks and veins, with some segmentation amongst the tiles. To improve the visual continuity of the results, it proposes a second iteration of the method to minimise segmentation. It achieves the initial promise of proposing a framework to synthesise material blocks for marble-like material with no size restrictions despite the quality of the results, which can be improved in future work. In addition, it can be used to translate a 2D image into a 3D model. As an example, a naturally flat-leaf image given as an exemplar can create a block of leaf-like material, where the underlying logic of a leaf is adapted into a material block. In the future, the method can be transformed into a material-generation tool to accomplish different material behaviours. If the method described in this paper will be used to generate metamaterials, various densities can be assigned to the clusters of generated geometries while generating blocks like the

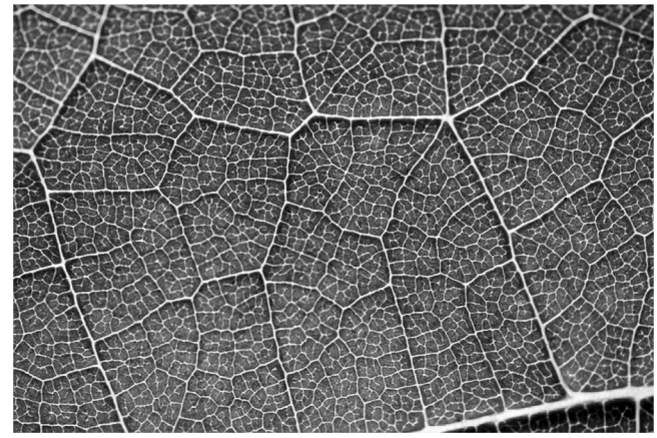

Fig. 23. Selected close-up image of a leaf.

Fig. 21. The sequence of marble tiles.

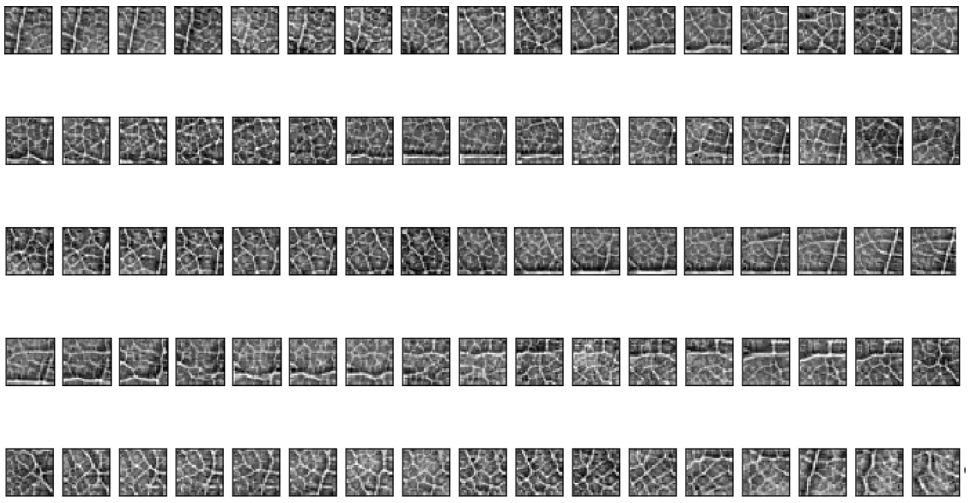

Fig. 24. The sequence of a leaf block generated with the EDCGAN.

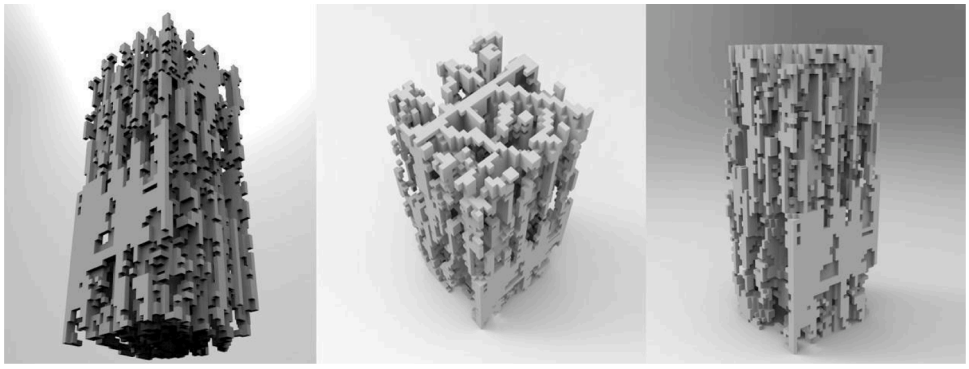

Fig. 25. The leaf block generated using EDCGAN. The grid originating from the leaf structure is gradually changing on the vertical axis.

Fig. 26. The 3D printed model of the generated leaf block.

images provided. The individual properties of these groups can be optimised based on their structural performance to maximise the overall efficiency.

For future work, the application field of the approach can be expanded to the prediction of hidden areas of material blocks which are not available in a scan due to resolution or penetration limitations. For this, the tools described in this paper should be modified to produce results closer to the real which can potentially be achieved by adding an interpolation step that merges various axes together. The number of exemplars should also be increased to provide other face images of a block for higher accuracy.

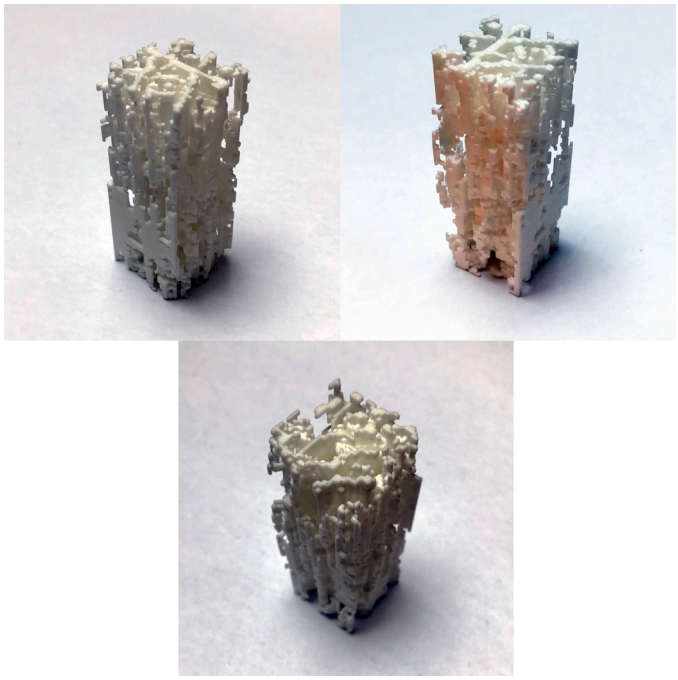

Fig. 27. The 3D printed model of the generated leaf block is stable and stands on its own.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] H. Nada, T. Nishimura, T. Sakamoto, T. Kato, Heterogeneous Growth of Calcite at Aragonite {001}- and Vaterite {001}-Melt Interfaces: a Molecular Dynamics Simulation Study, J. Cryst. Growth 450 (2016) 148–159, https://doi.org/10.1016/j.jcrysgro.2016.06.042.

[2] Bulakh, A., and H.-.R. Wenk. 2016. "Minerals: their Constitution and Origin". Cambridge UP.

[3] Pietroni, N., P. Cignoni, M.A. Otaduy, and R. Scopigno. 2007. "A Survey on Solid Texture Synthesis" 6, no. 1: 14.

[4] Cohn, D. 2009. "Photorealistic Rendering Techniques in AutoCAD®3D", 15.

[5] R. Mora, C. Bédard, H. Rivard, A Geometric Modelling Framework for Conceptual Structural Design from Early Digital Architectural Models, *Adv. Eng. Inf.*, Netw. Methods Eng. 22 (2) (2008) 254–270, https://doi.org/10.1016/j.aei.2007.03.003.

[6] W.J. Mitchell, A computational view of design creativity. Modeling Creativity and Knowledge-Base Creative Design, 1993, pp. 25–42.

[7] M.A. Zboinska, Influence of a Hybrid Digital Toolset on the Creative Behaviours of Designers in Early-Stage Design, J. Comput. Des. Eng. 6 (4) (2019) 675–692, https://doi.org/10.1016/j.jcde.2018.12.002.

[8] R.S. Kshetrimayum, A Brief Intro to Metamaterials, IEEE Potentials 23 (5) (2005) 44–46, https://doi.org/10.1109/MP.2005.1368916.

[9] Goodfellow, IJ., J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. 2014. "Generative Adversarial Networks". http://arxiv.org/abs/1406.2661.

[10] Radford, A., L. Metz, S. Chintala. 2015. "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks". November. http://arxiv.org/abs/1511.06434.

[11] Wei, L.-Y. and M. Levoy. 2000. "Fast Texture Synthesis Using Tree-Structured Vector Quantization". In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques – SIGGRAPH '00*. 479–88. https://doi.org/10.1145/344779.345009.

[12] S. Lefebvre, H. Hoppe, Parallel controllable texture synthesis, ACM Trans. Graph. 24 (3) (2005) 777–786.

[13] Bergmann, U., N. Jetchev, and R. Vollgraf. 2017. "Learning Texture Manifolds with the Periodic Spatial GAN".https://arxiv.org/abs/1705.06566.

[14] Perlin, K. 1985. "An image synthesizer". In Proceedings of the 12th annual conference on Computer graphics and interactive techniques (SIGGRAPH 85). Association for Computer Machinery, New York, NY, USA, 287–296. DOI: 10.1145/325334.325247.

[15] J. Kopf, C.-.W. Fu, D. Cohen-Or, O. Deussen, D. Lischinski, T.-.T. Wong, Solid Texture Synthesis from 2D Exemplars, ACM Trans. Graph. (2007), https://doi.org/10.1145/1275808.1276380.

[16] Zhang, H., T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. 2016. "StackGAN: text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks". December. https://arxiv.org/abs/1612.03242.

[17] Dosovitskiy, A., J.T. Springenberg, M. Tatarchenko, and T. Brox. 2014. "Learning to Generate Chairs, Tables and Cars with Convolutional Networks". *ArXiv: 1411.5928 [Cs]*. http://arxiv.org/abs/1411.5928.

[18] Zhao, X., J. Guo, L. Wang, F. Li, J. Zheng, and Bo Yang. 2021. "Solid Texture Synthesis Using Generative Adversarial Networks". *ArXiv:2102.03973 [Cs, Eess]*, https://arxiv.org/abs/2102.03973.

[19] Portenier, T., S. Bigdeli, and O. Goksel. 2020. "GramGAN: deep 3D Texture Synthesis From 2D Exemplars", *ArXiv:2006.16112 [Cs]*. http://arxiv.org/abs/2006.16112.

[20] J. Gutierrez, J. Rabin, B. Galerne, T. Hurtut, On Demand Solid Texture Synthesis Using Deep 3D Networks, Comput. Graph. Forum (2020), https://doi.org/10.1111/cgf.13889.

[21] R. Bostanabad, Reconstruction of 3D Microstructures from 2D Images via Transfer Learning, Comput.-Aided Des. 128 (2020), 102906, https://doi.org/10.1016/j.cad.2020.102906.

[22] Zhou, Y., Z. Zhu, X. Bai, D. Lischinski, D. Cohen-Or, and H. Huang. 2018. "Non-Stationary Texture Synthesis by Adversarial Expansion". *ArXiv:1805.04487 [Cs]*. https://arxiv.org/abs/1805.04487.

[23] Chen, H., J. Liu, W. Chen, S. Liu, and Y. Zhao. 2022. "Exemplar-Based Pattern Synthesis with Implicit Periodic Field Network". *ArXiv:2204.01671 [Cs]*. http://arxiv.org/abs/2204.01671.

[24] Henzler, P., N. J. Mitra, and T. Ritschel. 2020. "Learning a Neural 3D Texture Space From 2D Exemplars". In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 8353–61. Seattle. 10.1109/CVPR42600.2020.00838.

[25] W.T. Freeman, T.R. Jones, E.C. Pasztor, Example-Based Super-Resolution, IEEE Comput. Graph. Appl. 22 (2) (2002) 56–65, https://doi.org/10.1109/38.988747.

[26] Freeman, W.T. and E.C. Pasztor. 1999. "Learning Low-Level Vision". In *Proceedings of the Seventh IEEE International Conference on Computer Vision* (Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece: IEEE), 1182–89, vol. 2, https://doi.org/10.1109/ICCV.1999.790414.

[27] M.A. Tanner, W.H. Wong, The Calculation of Posterior Distributions by Data Augmentation, J. Am. Stat. Assoc. 82 (398) (1987) 528–540, https://doi.org/10.2307/2289457.

[28] Antoniou, A., A. Storkey, and H. Edwards. 2018. "Data Augmentation Generative Adversarial Networks". *ArXiv:1711.04340 [Cs, Stat]*, http://arxiv.org/abs/1711.04340.

[29] T.R. Shaham, T. Dekel, T. Michaeli, SinGAN: learning a Generative Model From a Single Natural Image, in: IEEE/CVF International Conference on Computer Vision (ICCV), IEEE, Seoul, Korea (South), 2019, pp. 4569–4579, https://doi.org/10.1109/ICCV.2019.00467, 2019.

[30] Shocher, A., S. Bagon, P. Isola, and M. Irani. 2019. "InGAN: capturing and Remapping the 'DNA' of a Natural Image". *ArXiv:1812.00231 [Cs]*. http://arxiv.org/abs/1812.00231.

[31] E. Linder-Norén, Keras Implementations of Generative Adversarial Networks: Eriklindernoren/Keras-GAN, Python, 2018. https://github.com/eriklindernoren/Keras-GAN.

[32] N. Herz, M. Waelkens, Classical Marble: Geochemistry, Technology, Trade, Brill Archive, 1988.

[33] S. Wolfram, A New Kind of Science, Wolfram Media Inc, First Edition, Champaign, IL, 2002.

[34] D. Raabe, Cellular Automata in Materials Science with Particular Reference to Recrystallization Simulation, Annu. Rev. Mater. Res. 32 (1) (2002) 53–76, https://doi.org/10.1146/annurev.matsci.32.090601.152855.

[35] P. Kearey, The Penguin Dictionary of Geology, 2nd edition, Penguin, London, 2001.

[36] Kingma, D.P., and M. Welling. 2014. "Auto-Encoding Variational Bayes". https://arxiv.org/abs/1312.6114.

[37] Yu, X., X. Zhang, Y. Cao, and M. Xia. 2019. "VAEGAN: a Collaborative Filtering Framework Based on Adversarial Variational Autoencoders". In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, 4206–12. Macao, China. 10.24963/ijcai.2019/584.

[38] Nguyen, A., A. Dosovitskiy, J. Yosinski, T. Brox, and J. Clune. 2016. "Synthesizing the Preferred Inputs for Neurons in Neural Networks via Deep Generator Networks". *ArXiv:1605.09304 [Cs]*. https://arxiv.org/abs/1605.09304.

[39] Zeiler, M.D., R. Fergus. 2014. "Visualizing and Understanding Convolutional Networks". In *Computer Vision – ECCV 2014*, edited by D Fleet, 8689:818–33. Cham: Springer International Publishing. 10.1007/978-3-319-10590-1_53.

[40] A.F. Cooper, D.L. Reid, Textural Evidence for Calcite Carbonatite Magmas, Dicker Willem, Southwest Namibia, Geology 19 (12) (1991) 1193–1196, https://doi.org/10.1130/0091-7613(1991)019≤1193:TEFCCM≥2.3.CO;2.

[41] L. Sack, C. Scoffoni, Leaf Venation: structure, Function, Development, Evolution, Ecology and Applications in the Past, Present and Future, New Phytologist 198 (4) (2013) 983–1000, https://doi.org/10.1111/nph.12253.