

# ORTHOIMAGE-TO-2D ARCHITECTURAL DRAWING WITH CONDITIONAL ADVERSARIAL NETWORKS

P. Agrafiotis <sup>1\*</sup>, G. Talaveros <sup>1</sup>, A. Georgopoulos <sup>1</sup>

<sup>1</sup>Lab. of Photogrammetry, School of Rural, Surveying and Geoinformatics Eng., National Technical University of Athens, Greece  
(pagraf, drag)@central.ntua.gr, giwrgostas@gmail.com

**KEY WORDS:** Orthoimage, 2D architectural drawing, Conditional GANs, Edge Detection, Vectorization

## ABSTRACT:

Vectorization of orthoimages of Cultural Heritage sites requires a considerable amount of time and constant supervision by qualified professionals. In addition, this 2D architectural drawing creation requires expert knowledge for appropriate interpretation of the orthoimages. In this paper, the use of conditional adversarial networks as a solution to orthoimage-to-drawing translation problems is proposed. The presented work exploits a state of the art conditional Generative Adversarial Network with a Markovian discriminator and modifies it using a ResNet fully convolutional network as generator in order to deliver reliable and accurate 2D architectural drawings in a binary image format. Following the 2D drawing image generation, their automated conversion into vector files is performed through a vectorization function, giving also the possibility to edit and scale the edges. Experimental results over two different Cultural Heritage test sites demonstrates that this approach is highly effective at synthesising 2D architectural drawings from orthoimages in great detail and reliability by learning the interpretation performed by the expert architects during the vectorization process.

## 1. INTRODUCTION

For centuries, construction or indeed conservation works were based on drawings, architectural or technical drawings on paper or other flat surfaces. 2D architectural drawings are important because they are used to communicate the technical details of a project or a construction in a common and universally understandable format. Up to now, drawings are printed, mainly on paper sturdy enough to be handled outdoors and be resistant to natural phenomena. In recent years however, digital technology has advanced remarkably and consequently all drawings, 2D or even 3D, are produced and disseminated in digital form. Either in vector or raster format, 2D drawings can nowadays be displayed on flat screens of laptops, tablets or even smartphones. However, this display does not fully appeal to field experts as expected. Digital screens are not displaying correctly under sunlight and are difficult to handle and make quick notes, although theoretically all pertinent tools are there. The primary role or function of working drawings is to convert design data into construction information and to clearly communicate that information to building industry, code officials, product manufacturers, suppliers, and fabricators. Keeping track of modifications and/or additions during documentation is a necessary step in getting an accurate drawing set for the working system. These drawings are a valuable resource for maintenance and troubleshooting. The drawings, however, must be maintained after documentation if they are to continue to be useful. Hence, 2D drawings are still necessary and required, as by creating a drawing than simply presenting captured 3D data set an interpretation of the subject can provide:

- Accessible, platform independent, information.
- Reliable perception of scale, a printed plot, either dimensioned or with a scale, allows consistent shared experience of information.

- Completeness with added structural detail in context.
- Simplicity by showing selected information pertinent to a given project.
- Clarification of complex 3D spaces by use of plan, section or perspective views.
- Architectural understanding by showing forms clearly as they relate to style or typologies without distortion.
- Legally immutable records in cases such as planning applications and construction contracts<sup>1</sup>.

Finally, completeness in a drawing requires careful examination of the structure not covered by surface recording methods like photogrammetry or laser scanning. Automated architectural drawing producing approaches shrink the laborious process of manually producing the vector drawings and provide ample time for in situ control, updating, and understanding of structure. Such an automated approach is presented and evaluated in this paper by exploiting conditional adversarial networks as a solution to orthoimage-to-drawing translation problems. Following the 2D drawing image generation, their automated conversion into vector files is performed through a vectorization function. The rest of the paper is structured as follows: Section 2 presents a brief review of the literature. Section 3 describes the proposed methodology while section 4 presents and analyses the performed experiments. Section 5 comments on the results of the method and concludes the paper.

## 2. RELATED WORK

Considering that the translation of an orthoimage into 2D drawing falls into the general problem of edge detection, a brief

---

\* Corresponding author

<sup>1</sup><https://billboyheritagesurvey.wordpress.com/2022/09/23/on-draughtsmanship-and-the-2and-a-half-d-world/>

overview of the related work on 2D edge detection as well as in conditional GANs is given next.

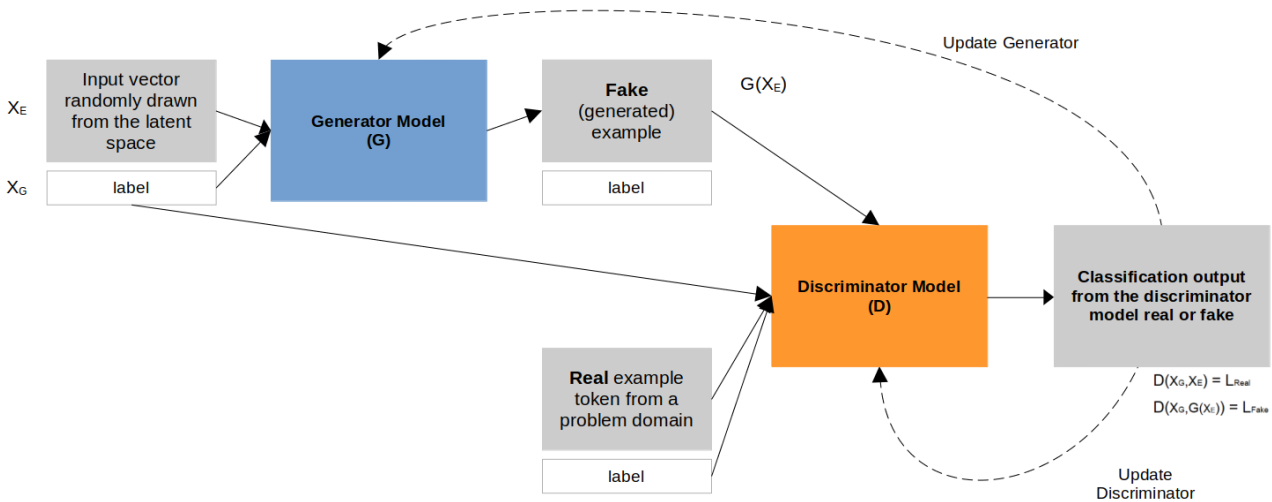
## 2.1 2D Edge Detection

Significant local changes in the image intensity, usually associated with a discontinuity in either the image intensity or the first derivative of the image intensity are called edges and are important features for image analysis. They typically occur on the boundary between two different regions in an image and can be defined as a set of connected pixels that form a boundary between two regions. Due to its importance, edge and contour detection and mainly 2D edge detection is a fundamental computer vision problem. Early approaches on edge detection, such as Sobel (Sobel et al., 1968), Prewitt (Prewitt et al., 1970), Scharr (Kroon, 2009), Kirsh (Kirsch, 1971), MarrHildreth (Marr and Hildreth, 1980), and Canny (Canny, 1986) used handcrafted features. Most recently, convolutional neural networks (CNNs) have been applied to the edge detection problem, delivering better results compared to the traditional approaches (Xie and Tu, 2015; Poma et al., 2020; Su et al., 2021). A major success of this kind of work is Holistically-Nested Edge Detection (HED) (Xie

and Tu, 2015), a CNN model that achieves near-human edge detection accuracy on standard datasets. In recent years, automatic feature learning by CNNs has replaced explicit edge detection for higher-level vision tasks like image classification (Cosgrove and Yuille, 2020). However, it is well known that CNNs learn edge-like features implicitly (Krizhevsky et al., 2012).

## 2.2 Conditional GANs

GANs have been vigorously studied in the last years, especially to formulate image synthesis problems. Earlier papers have focused on specific applications, and till 2017 it has remained unclear how effective image-conditional GANs can be as a general-purpose solution for image-to-image translation. Conditional GANs (Figure 1) are different in that the loss is learned, and can, in theory, penalise any possible structure that differs between output and target (Goodfellow et al., 2020, Mirza and Osindero, 2014, Isola et al., 2017). This way, conditional image synthesis allows users to use their creative inputs to control the output of image synthesis methods.



**Figure 1.** Conditional GAN (cGAN) model architecture followed in this work. The discriminator, D, learns to classify between fake (synthesised by the generator) and real [image, drawing] tuples. The generator, G, learns to fool the discriminator. Unlike an unconditional GAN, both the generator and discriminator observe the input label.

## 3. METHODOLOGY

Instead of exploiting edge detection algorithms, the methodology presented is based on a conditional GAN (Figure 1), since it would fit better to orthoimage-to-drawing translation problems such as the one discussed in this paper, by learning the necessary generalisation-interpretation performed when vectorizing orthoimages of Cultural Heritage sites by qualified professionals.

In the same problem, an edge detector would deliver all the visible edges which is out of the scope of this work. In the literature, conditional GANs are being used mostly for edges to photo translation (Isola et al., 2017) and not the opposite direction, as performed in this paper. Following cGAN results, processing of the generated 2D drawing images is finished by converting them from visual data, to vector, through a function to convert raster information to vector.

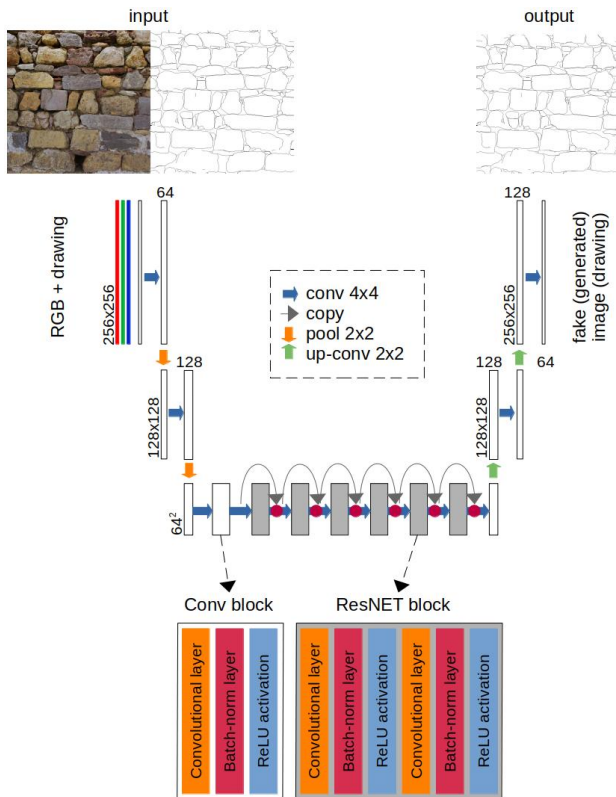
### 3.1 Conditional GAN

GANs are generative models that learn a mapping from random noise vector  $z$  to output image  $y$ ,  $G : z \rightarrow y$  (Goodfellow et al., 2020). In contrast, conditional GANs (cGANs) learn a mapping from observed image  $x$  and random noise vector  $z$ , to  $y$ ,  $G : \{x, z\} \rightarrow y$ . The generator (G) is trained to synthesise outputs that cannot be distinguished from “real” images by an adversarially trained discriminator, D, which is trained to do as well as possible at detecting the generator’s “fakes” (Isola et al., 2017). This procedure is illustrated in Figure 1. In the following sections details about the Generator and Discriminator networks used in this work are given.

#### 3.1.1 Generator

The methodology adopted in this work is based on the Pix2Pix (Isola et al., 2017) framework. However, instead of using “U-Net”-based architecture (Ronneberger et al., 2015) as a generator, as in the original presented configuration of Pix2Pix, a ResNet-

based architecture (He et al., 2016) with 6 convolutional blocks with skip connections is exploited here, adapted from Johnson et al., (2016). This change was decided after performing various configurations and extensive tests for this specific application, demonstrating that the ResNet-based architecture delivered more detailed and less noisy outputs. This ResNet architecture is implemented also in recent popular GAN models (Karras et al., 2018, Zhu et al., 2017, Miyato et al., 2018, Zhang et al., 2019). As shown in Figure 2, the generator comprises six residual blocks, and nearest neighbour up-sampling layers. Each residual block contains two convolutional layers with the skip connection, and a learned residue of input is added to the output to ensure the characteristics of original features are retained (Xia et al., 2021).

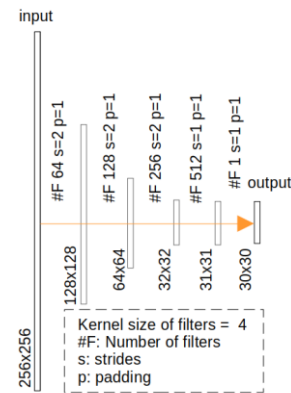


**Figure 2.** The ResNet-based generator architecture for an input image of 256x256 size.

### 3.1.2 Markovian Discriminator

For the discriminator, a convolutional "PatchGAN" classifier is used, as in the original Pix2Pix framework, which penalises structure at the scale of 70x70 pixels image patches. This way, local statistics regarding the style of the input image will be kept by the network, leading to a more realistic output. Specifically, PatchGAN, after feeding one input image to the network, provides the probabilities of each  $N \times N$  patch in an image being real or fake, but not in a scalar output. Here  $N \times N$  can be different depending on the dimension of an input image (it is 30x30 for a 256x256 image, see Figure 3), but each output vector represents 70x70 patches of an input image. The discriminator runs convolutionally across the image, averaging all responses to provide the ultimate output of D. In Isola et al., (2017), it is demonstrated that N can be much smaller than the full size of the image and still produce high quality results. This smaller PatchGAN has fewer parameters, runs faster, and can be applied to arbitrarily large images. Such a discriminator effectively

models the image as a Markov random field, assuming independence between pixels separated by more than a patch diameter (Isola et al., 2017). Therefore, this PatchGAN can be understood as a form of texture/style loss for maintaining the style of the ground truth architectural drawing (Isola et al., 2017).



**Figure 3.** Discriminator (PatchGAN) architecture with 70x70 patches of an input image of 256x256 size.

### 3.1.3 Optimization and Inference

To optimise those two networks, training proceeds in alternating periods as in Goodfellow et al., (2020). One gradient descent step on the discriminator is followed by one step on the generator. During the discriminator training phase, the generator is kept constant. Similarly, we keep the discriminator constant during the generator training phase. Otherwise the generator would be trying to hit a moving target and might never converge. As suggested in the original GAN paper (Goodfellow et al., 2020), rather than training the generator to minimise  $\log(1 - D(x, G(x, z)))$ , it is instead trained to maximise  $\log D(x, G(x, z))$  (Goodfellow et al., 2020). In addition, objective (Equation 1) is divided by 2 while optimising the discriminator, which slows down the rate at which it learns relative to the generator. Mini batch SGD is used and the Adam solver (Kingma and Ba, 2014) is applied with a learning rate of 0.0002, and momentum parameters  $\beta_1 = 0.5$ ,  $\beta_2 = 0.999$ . In inference, the generator network runs in exactly the same manner as during the training phase. Batch normalisation (Ioffe and Szegedy, 2015) is also applied using the statistics of the test batch, rather than aggregated statistics of the training batch (Isola et al., 2017). This approach to batch normalisation, when the batch size is set to 1, has been demonstrated to be effective at image generation tasks in the literature (Ulyanov et al., 2016). For the performed experiments, a batch size of 1 is used while to retrieve the best optimization parameters, 69 different experimental setups were set and results were evaluated.

As in Isola et al., (2017) the optimisation of the G and D can be expressed as:

$$L_{dGAN}(G, D) = E_{(x_G, x_E)} [\log D(x_G, x_E)] + E_{(x_G, G(x_E))} [\log [1 - D(x_G, G(x_E))]] \quad (1)$$

In order for the generator to be near the ground truth output L1 is used:

$$L_{L1}(G) = E_{(x_G, x_E)} [\|D(x_G, x_E) - G(x_E)\|_1] \quad (2)$$

The final objective is:

$$G_{Final} = \arg \min_G \max_D L_{dGAN}(G, D) + \lambda * L_{L1}(G) \quad (3)$$

Where:

$$D(X_G, X_E) = L_{Real} \quad (4)$$

$$D(X_G, G(X_E)) = L_{Fake} \quad (5)$$

$$E_{x_c, x_e}(\log[D(X_G, X_E)]) = \text{logarithmic value of } L_{Real} \quad (6)$$

$$E_{x_c, G(x_e)}(\log[1 - D(X_G, G(X_E))]) = \text{logarithmic value of } L_{Fake} \quad (7)$$

### 3.2 Binary Drawing Image to Vector

Following the 2D drawing image generation by the cGAN which depicts the boundaries of the stones in black and stone surface in white, this stage achieves their conversion into vector files, which gives the possibility to edit and scale the edges. To this end, a function which converts the image information to a vector is implemented. This function creates vector polygons for all connected regions of pixels in the raster sharing a common pixel value, and specifically the 0 value for this paper. The function calculates the maximum distance between two connectable pixels, based on the hypothesis that the pixel height  $\Delta y$  and width  $\Delta x$  are the same. Consequently gets the trajectory of the pixels that have value equal to 0 and saves those coordinates as points. Each polygon is created with an attribute indicating the pixel value of that polygon. A raster mask may also be provided to determine which pixels are eligible for processing.

## 4. EXPERIMENTS

To explore the potential of the proposed approach the method is tested over two totally different Cultural Heritage test sites, forming the respective datasets described below.

### 4.1 Test sites and datasets

To demonstrate the effectiveness of the proposed methodology in synthesising 2D architectural drawings from orthoimages as well as investigate possible drawbacks and limitations, data from two different Cultural Heritage test sites with masonry walls were used. Specifically, orthoimages and drawings from Chios Fortress and Dafni Monastery were exploited. Those two sites are not only different in terms of construction era, masonry walls' style and construction technique, but also in terms of data processing pipeline, enabling the demonstration of the generalisation capabilities of the approach in different domains (masonry walls, roof tiles etc.).

#### 4.1.1 Chios Fortress

Chios is an island in the Eastern part of the Aegean Sea, approximately 220 km eastwards from Athens. The Medieval fortress of Chios covers an area of 180,000 m<sup>2</sup> and nowadays its walls include a residential complex with 650 inhabitants. It used to be the walled core of the town since the Genovese period, however there is evidence that the Castle had been inhabited since the Hellenistic years (4<sup>th</sup> c. BC), as well as during the Roman period and the early Byzantine years. Its walls both in land and at sea form an irregular pentagon, with strong bastions, eight of which are still preserved. For the documentation of the parts of interest, it was decided to employ close-range automated photogrammetry and image-based modelling, as well as terrestrial laser scanning and topographic surveys. In particular, the following equipment were used: An Integrated Total Station (Topcon GPT3003LN), a time-of-flight pulse-based 3D Laser Scanner (Leica Scanstation 10), two DSLR cameras (Canon EOS

6D and Canon 80D with 8-15mm, 24mm and 18-55mm Lenses), and an Unmanned Aerial Vehicle (DJI Phantom 4 Pro). The 3D detailed textured model as well as the orthophotos of planar and cylindrical surfaces were generated through Agisoft Photoscan v.1.4.2. The overall process in this software, as is the case with all similar ones, involves contemporary computer vision algorithms (Structure from Motion and Multi-View Stereo) adapted to confront the challenge of processing a huge number of images and extracting useful metric information from them. The orthophotos and the 2D drawings that were produced with a GSD of 2mm, among other, include the façade of the walls at scales 1:25 and 1:40 (Figure 4) which is the one used in this paper. The production of the required architectural vector drawings was mainly based on the orthophotos. These drawings were produced by suitably tracing the orthophotos in a CAD environment, in order to represent the masonry, the structural details and the pathology. The initial digital images were always available to be used for proper interpretation in dubious cases.

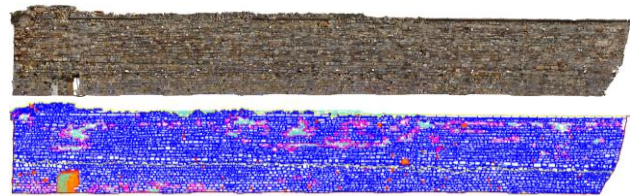


Figure 4. Up: Orthoimage of Chios Fortress in orthoimage (above) and the respective 2D architectural drawing (below).

#### 4.1.2 Dafni Monastery

The Dafni Monastery is one of the two remaining excellent specimens of the culmination of Byzantine architecture (Figure 5). It was built in the 11th century and is situated in the southeastern part of Attica near Athens.

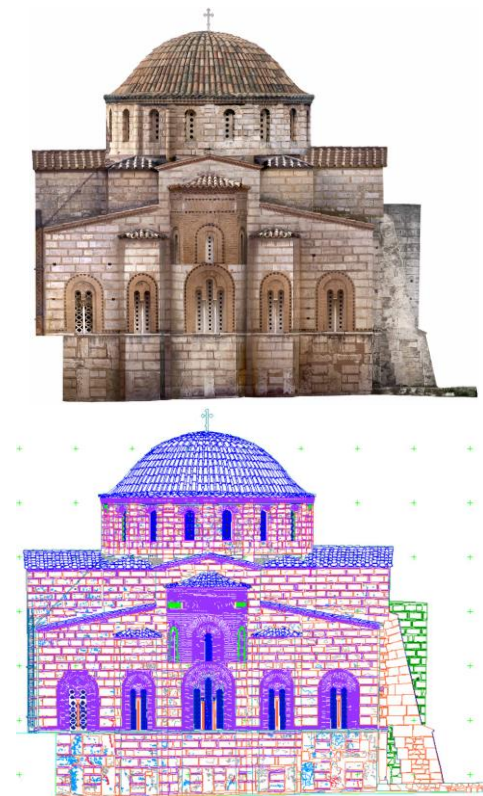


Figure 5. The east facade of Dafni site in orthoimage (above) and the respective 2D architectural drawing (below).

The whole monastery extends on an area of 0.7 hectares and in the centre of that area lies the majestic central church, the Katholikon. In essence it is a cross-domed octagon type of church extending approximately 25 x 15 m<sup>2</sup> and 20 m in height. The Monastery is considered to be the Parthenon of the Byzantine era and is internationally protected by UNESCO.

A complete and thorough geometric recording of the monument has been carried out (Georgopoulos et al. 2004). In addition it was decided to produce a 3D digital model based on the aforementioned survey. The 3D rendering could make use of all conventional survey measurements and digital photogrammetric products. These products were either raster orthoimages or vector drawings. The fieldwork covered approximately 30% of this time and for the purposes of the project around 11500 points were measured and 1000 photographs were taken using a Zeiss UMK1318 metric camera and a 6x6 analogue Hasselblad camera. Digital photogrammetric techniques, mostly performed on a Z/I SSK workstation, were employed for the production of the final products. Among the final products were orthophotomosaics at a scale of 1:25 with a GSD of 2mm and corresponding vector drawings produced by experts within the AutoCAD environment (Figure 5).

#### 4.1.3 Generating Pairs for Training and Testing

To facilitate training and testing of the conditional GAN, consecutive non-overlapping patches of 1024x1024 pixels were extracted by the above data and training and testing data were generated in the form of pairs of images (Figure 6).



**Figure 6.** Example of 1024x1024 data creation. The pair consists of the orthoimage (left) and the 2D drawing made by an expert architect (right). The first row depicts a pair from Chios Fortress while the second row a pair from Dafni Monastery, and specifically the roof tiles. 84 pairs were generated for the Chios Fortress dataset in total while for the Dafni Monasteri 33 pairs.

## 4.2 Experimental Results

To demonstrate the capabilities of the proposed methodology as well as identify any possible limitations, seven different training and testing approaches with different configurations were performed (Table 1). This way, the amount of the necessary data per dataset as well as the domain adaptation capabilities were investigated.

The rationale behind training only on the 20% of the available data is that when having a big amount of data to vectorize, an expert could vectorize some of these and let the network vectorize the rest. The rationale behind training in one site and testing in another is to evaluate the domain adaptation and the generalisation possibilities of the trained networks: having a

model trained on one site, vectorizing an orthoimage of another site, without fine tuning the model on the target site.

Approach #	Data for training % of images used	Data for testing % of images used
1	Chios 80%	Chios 20%
2	Chios 50%	Chios 50%
3	Chios 20%	Chios 80%
4	Chios 100%	Dafni 100%
5	Dafni 100%	Chios 100%
6	Dafni 80%	Dafni 20%
7	Chios+Dafni 80%	Chios+Dafni 20%

**Table 1.** The different test and training approaches performed.

In the figures below, results of the most characteristic training and testing approaches are presented in order to illustrate the accuracy of the trained models as well as the cases that they slightly fail due to special characteristics of the input imagery.

#### 4.2.1 Train on 20% and test on 80% of Chios dataset

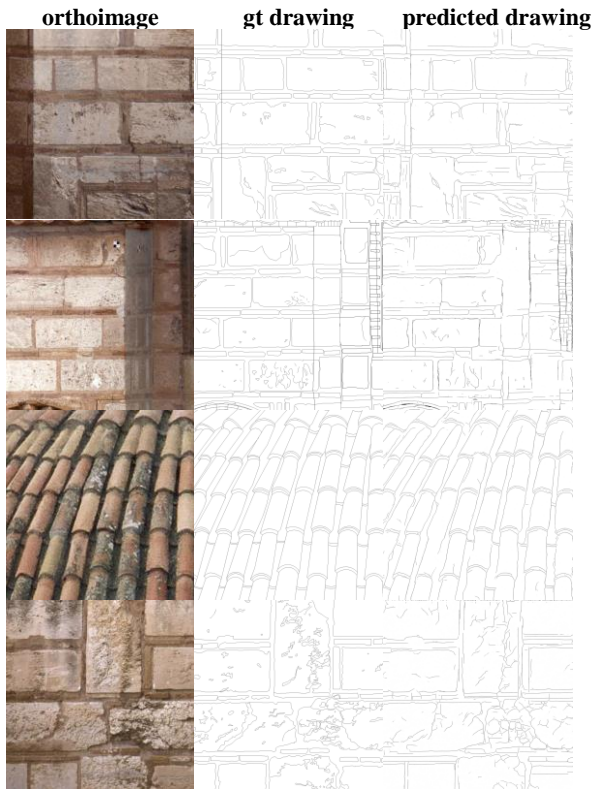
In this section, results over the test site of Chios are presented. Specifically, to train the model, 20% of the images were used while testing is performed on the rest 80% of the data, being unseen by the model during training.



**Figure 7.** Example results on Chios dataset. The left column depicts the RGB image, the middle column depicts the ground truth (gt) 2D drawing, and the right column depicts the predicted 2D drawing vectorized.

#### 4.2.2 Train on 20% and test on 80% of Dafni dataset

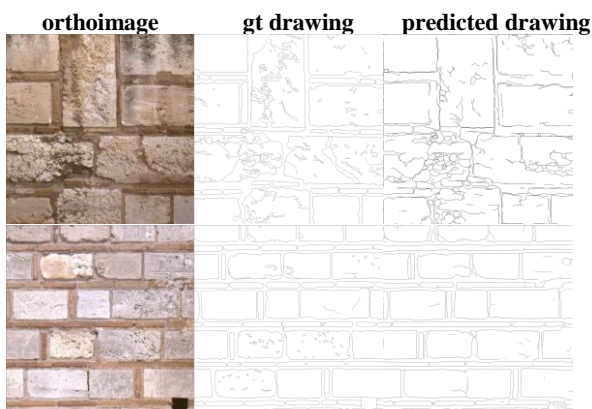
In this section, results over the test site of Dafni are presented. Specifically, to train the model, 20% of the images were used while testing is performed on the rest 80% of the data, being unseen by the model during training.



**Figure 8.** Example results on Dafni dataset. The left column depicts the RGB image, the middle column depicts the ground truth (gt) 2D drawing, and the right column depicts the predicted 2D drawing vectorized.

#### 4.2.3 Train on Chios and test on Dafni dataset

In this section, results over the test site of Dafni are presented. Contrary to the aforementioned approaches, to train the model here, 100% of the Chios images were used while testing is performed on the Dafni data, being unseen by the model during training, in order to demonstrate and evaluate the generalisation potential of the model.

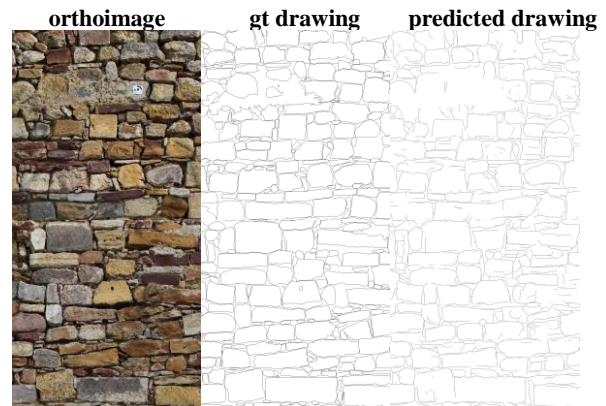


**Figure 9.** Example results on Dafni dataset. The left column depicts the RGB image, the middle column depicts the ground truth (gt) 2D drawing, and the right column depicts the predicted 2D drawing vectorized.

#### 4.2.4 Train on Dafni and test on Chios dataset

In this section, results over the test site of Chios are presented. Contrary to the aforementioned approaches, to train the model

here, 100% of the Dafni images were used while testing is performed on the Chios data, being unseen by the model during training, in order to demonstrate and evaluate the generalisation potential of the model once again.



**Figure 10.** Example results on Chios dataset. The left column depicts the RGB image, the middle column depicts the ground truth (gt) 2D drawing, and the right column depicts the predicted 2D drawing vectorized.

#### 4.2.5 Train on both Chios and Dafni and test on both



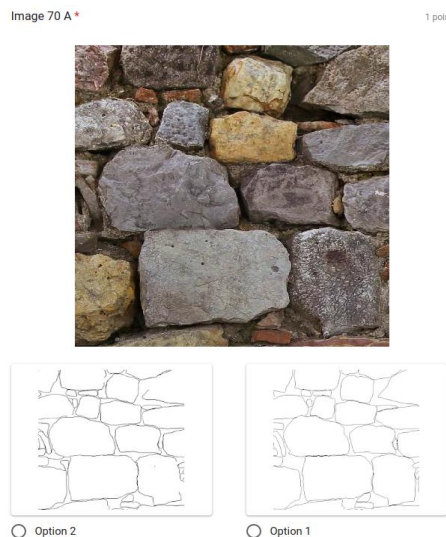
**Figure 11.** Example results on Dafni (first row) and Chios (second row) datasets. The left column depicts the RGB image, the middle column depicts the ground truth (gt) 2D drawing, and the right column depicts the predicted 2D drawing vectorized.

### 4.3 Visual Evaluation

Results of the above experiments suggest that the network achieves very realistic results, even when trained in one dataset and tested on another. Most common errors found here are in areas with plasters where distinguishing stones' edges is very difficult. Also, it is noticed that the network sometimes delivers edges from rock details and not their boundaries. Also, when the limits of the rocks were not clear enough, the network failed to deliver an edge, resulting in leaving some rocks with an "open" polyline. Surprisingly enough, results from the tests performed by training the model in one dataset and testing on another demonstrated the generalisation and domain adaptation possibilities, since the model worked even though it was trained in different sensor data, different architectural style and different characteristics (shape, colour etc.) of the stones.

#### 4.4 Questionnaire-based Evaluation

Except for the visual evaluation performed by the authors, evaluation was also performed by the end users and professionals i.e. architects and archaeologists. To that direction, a questionnaire was prepared and sent to 75 experts. Google Forms platform was used to create the questionnaire which consists of 15 different cases/questions where respondents should choose between the real drawing made by an architect and used as a ground truth for training the cGAN models and the image generated by the cGAN. Those two options were given randomly as possible selections to avoid biased replies. For each reply where the responder selected the real drawing, one point was given. So the closer the score to 0, the more realistic the cGAN generated images are. Accordingly, the closer the score is to 15, the more unrealistic the cGAN generated images are. Figure 12 depicts a case/question of the created questionnaire.



**Figure 12.** Example question of the created questionnaire.

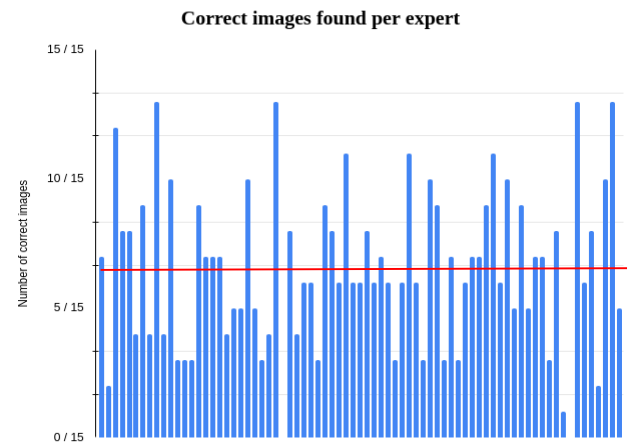
Above is the RGB image and below the respective expert-made and predicted 2D drawing. Responders had to select the choice they believe is the expert-made one.

In the y axis of Figure 13, the number of the correct responses collected per responder are presented. Additionally, the average score was calculated equal to 6.51/15 points, the median score 6/15 points, while the minimum and the maximum correct responses ranged between 0 and 13/15 points. Interestingly enough, those results suggest a confusion between distinguishing the real from synthetic drawing. Missing columns denoting 0 score.

This testifies the great similarity of the cGAN synthetic 2D drawings to the real ones. At the same time, there are also observed cases of completely successful discrimination against them. For those cases, this remark shows that the synthetic images are still not perfect and need improvement. However, there are also apparent cases of completely unsuccessful discrimination. It is worth noting that in 2 questions the synthetic drawing prevailed over the real one by 70.66% (53 responders).

Through the valuable discussion followed after the questionnaire, positive comments and conclusions were given. It was reported that synthetic 2D drawings are good enough and persuasive, making it difficult to distinguish between real and synthetic drawings. Also while completing the questionnaire, it was observed that the key element that showed the differentiation of

the two images wasn't the edge rendering part, but the "hidden edges" delineation behind occlusions or plaster, as also reported previously. Finally for similar future projects, the use of this process was considered extremely helpful since it offers acceptable results in repetitive segments that do not require special attention and potentially more information like cracks, damages or carvings, regarding the pathology of the depicted site, which is often an additional and very important purpose of Cultural Heritage documentation projects.



**Figure 13.** The number of the correct responses collected for each of the 75 responders. Red line represents the average score.

#### 5. CONCLUSIONS

Main goal of the research work presented in this paper was to generate accurate and realistic 2D drawings from orthoimages of Cultural Heritage masonry buildings. To achieve that, a conditional Generative Adversarial Network with a ResNet-based fully convolutional network as generator and a PatchGan as a discriminator were used. Extensive training and testing demonstrated the great potential of the proposed approach, while visual evaluation, questionnaire results and experts' feedback indicated that the objective of this work has been primarily achieved.

Indeed, using an orthoimage of a Cultural Heritage masonry construction, a 2D drawing of this orthoimage can be generated automatically and in a very short time in vector form. Surprisingly enough, results from the tests performed by training the model in one dataset and testing on another demonstrated high generalisation and domain adaptation possibilities, since the model worked even though it was trained in different sensor data, different architectural style and different characteristics (shape, colour etc.) of the stones. However, some minor flaws also exist. Most common errors found here are in areas with plasters where distinguishing stones' edges is very difficult. In cases like those, some edges were not detected, resulting in gaps and discontinuities of the 2D drawing. Also, the network faced difficulties in detecting only the outer edges of the stones, ignoring crack, carvings etc. However, this might be considered also as positive, since more information is produced that perhaps an expert will ignore. Proposed work can be very useful when having a big amount of data to vectorize; an expert could vectorize some of these and let the network vectorize the rest or even data from another site, without fine tuning the model, if possible.

## ACKNOWLEDGEMENTS

This work is part of the HYPERION project. HYPERION has received funding from the European Union's framework programme for Research and Innovation (HORIZON 2020) under Grant Agreement No. 821054. The content of this publication is the sole responsibility of NTUA and does not necessarily reflect the opinion of the European Union. For all figures permissions have been obtained from the owners.

## REFERENCES

- Canny, J. F., 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, 679–698.
- Cosgrove, C., & Yuille, A. (2020). Adversarial examples for edge detection: They exist, and they transfer. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (pp. 1070-1079).
- Georgopoulos, A., Ioannidis, Ch., Daskalopoulos, A., Demiridi, E., 2004. 3D Reconstruction and Rendering for a Virtual Visit, *International Congress of ISPRS, Istanbul, July 2004*, <http://www.isprs.org/proceedings/XXXV/congress/comm5/papers/629.pdf>
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., ... & Bengio, Y. (2020). Generative adversarial networks. *Communications of the ACM*, 63(11), 139-144.
- H. Zhang, I. Goodfellow, D. Metaxas, A. Odena Self-attention generative adversarial networks *International Conference on Machine Learning, PMLR (2019)*, pp. 7354-7363
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Ioffe, S., & Szegedy, C. (2015, June). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning* (pp. 448-456).
- Isola, P., Zhu, J. Y., Zhou, T., & Efros, A. A. (2017). Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1125-1134).
- J.-Y. Zhu, T. Park, P. Isola, A.A. Efros Unpaired image-to-image translation using cycle-consistent adversarial networks *Proceedings of the IEEE international conference on computer vision (2017)*, pp. 2223-2232
- Johnson, J., Alahi, A., & Fei-Fei, L. (2016). Perceptual losses for real-time style transfer and super-resolution. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part II 14* (pp. 694-711). Springer International Publishing.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kirsch, R. A., 1971. Computer determination of the constituent structure of biological images. *Computers and biomedical research*, 4(3), 315–328.
- Kroon, D., 2009. Numerical optimization of kernel based image derivatives. *Short Paper University Twente*, 3.
- Krizhevsky, A., Sutskever, I. and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- M. Mirza, S. Osindero Conditional generative adversarial nets *arXiv preprint arXiv:1411.1784 (2014)*
- Marr, D., Hildreth, E., 1980. Theory of edge detection. *Proceedings of the Royal Society of London. Series B. Biological Sciences*, 207(1167), 187–217.
- Poma, X. S., Riba, E., Sappa, A., 2020. Dense extreme inception network: Towards a robust cnn model for edge detection. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 1923–1932.
- Prewitt, J. M. et al., 1970. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1), 15–19.
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18* (pp. 234-241). Springer International Publishing.
- Sobel, I., Feldman, G. et al., 1968. A 3x3 isotropic gradient operator for image processing. a talk at the Stanford Artificial Project in, 271–272.
- Su, Z., Liu, W., Yu, Z., Hu, D., Liao, Q., Tian, Q., Pietikainen, M., Liu, L., 2021. Pixel difference networks for efficient edge detection. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 5117–5127.
- T. Karras, T. Aila, S. Laine, J. Lehtinen Progressive growing of GANs for improved quality, stability, and variation *International Conference on Learning Representations (ICLR) (2018)*
- Tapinaki, S., Skamantzari, M., Chliverou, R., Evgenikou, V., Konidi, A. M., Ioannatou, E., Mylonas, A., Georgopoulos A., 2019. 3D Image Based Geometric Documentation of a Medieval Fortress. *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLII-2/W9*, 699–705, <https://doi.org/10.5194/isprs-archives-XLII-2-W9-699-2019>
- Ulyanov, D., Vedaldi, A., & Lempitsky, V. (2016). Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*.
- Xia, Y., Ravikumar, N., Greenwood, J. P., Neubauer, S., Petersen, S. E., & Frangi, A. F. (2021). Super-resolution of cardiac MR cine imaging using conditional GANs and unsupervised transfer learning. *Medical Image Analysis*, 71, 102037.
- Xie, S., Tu, Z., 2015. Holistically-Nested Edge Detection. 9.