Synthetic data generation and testing for the semantic segmentation of heritage buildings

Eugénio Pellis, Andrea Masiero, Pierre Grussenmeyer, Michele Betti, Grazia Tucci

SYNTHETIC DATA GENERATION AND TESTING FOR THE SEMANTIC SEGMENTATION OF HERITAGE BUILDINGS

E. Pellis [1,2]*, A. Masiero[1], P. Grussenmeyer [2], M. Betti[1], G. Tucci [1]

[1] Department of Civil and Environmental Engineering (DICEA), University of Florence, 50139 Florence, Italy - (eugenio.pellis, andrea.masiero, grazia.tucci, michele.betti)@unifi.it
[2] Photogrammetry and Geomatics Group, ICube Laboratory UMR 7357, CNRS, INSA Strasbourg, Université de Strasbourg, 67000 Strasbourg, France - pierre.grussenmeyer@insa-strasbourg.fr

KEY WORDS: synthetic data, image semantic segmentation, deep learning, heritage buildings

ABSTRACT:

Over the past decade, the use of machine learning and deep learning algorithms to support 3D semantic segmentation of point clouds has significantly increased, and their impressive results has led to the application of such algorithms for the semantic modeling of heritage buildings. Nevertheless, such applications still face several significant challenges, caused in particular by the high number of training data required during training, by the lack of specific data in the heritage building scenarios, and by the time-consuming operations to data collection and annotation. This paper aims to address these challenges by proposing a workflow for synthetic image data generation in heritage building scenarios. Specifically, the procedure allows for the generation of multiple rendered images from various viewpoints based on a 3D model of a building. Additionally, it enables the generation of per-pixel segmentation maps associated with these images. In the first part, the procedure is tested by generating a synthetic simulation of a real-world scenario using the case study of Spedale del Ceppo. In the second part, several experiments are conducted to assess the impact of synthetic data during training. Specifically, three neural network architectures are trained using the generated synthetic images, and their performance in predicting the corresponding real scenarios is evaluated.

1. INTRODUCTION

Over the past few years, machine learning (ML) and deep learning (DL) techniques have gained popularity for tasks such as image classification, semantic segmentation, and object detection. One promising application in this domain is the generation of 3D Building Information Models (BIM) using deep learning-based semantic segmentation, known as Scan-To-BIM. However, training such models requires large datasets of supervised examples, which can be time-consuming to collect and process accurately. These challenges are especially prevalent in complex scenarios that demand a diverse range of finely annotated training samples to enhance model generalization and capability. To address these limitations, the use of synthetic data has emerged as a common approach. Synthetic data refers to artificially generated data that imitate real-world observations, enabling the training of machine learning algorithms when actual data collection is difficult or costly. Synthetic datasets can include binary, numerical, categorical, or unstructured data, such as images or videos. Synthetic data offers several advantages, including customization, cost-effectiveness, quick production, and data privacy preservation. However, generating synthetic data is still a complex process that requires skilled operators. If not appropriately synthesized, the data may provide an inaccurate representation of real-world events, leading to biases in the obtained results. Inaccurate or misrepresented synthetic data can hinder the proper testing and training of machine learning systems, as they fail to capture the essential patterns required for accurate performance. This work focuses on the use of synthetic data for semantic segmentation in heritage building scenarios. Specifically, we present a workflow for creating synthetic rendered image data to train a multiview-based deep learning classifier proposed in the work by Pellis et al. (2022). The paper outlines a process to generate highly accurate rendered images from a Building Information Modeling (BIM) or three-dimensional (3D) model of a historical building, with a case study conducted on the Spedale del Ceppo in Pistoia, Italy. Additionally, several training tests were conducted to evaluate the impact of synthetic data during the training process.

2. EXISTING APPROACHES

Generating synthetic data requires a robust model capable of recreating realistic datasets based on specific features of the target data. There are four main categories of methods used for synthetic data generation:

Variational Autoencoders (VAEs) are autoencoders that incorporate regularization in the training process to ensure that the latent space possesses desirable properties for generating accurate new data. During training, a "reconstruction error" is computed and minimized by the model. VAEs are effective for continuous data but less so for categorical data.

Generative Adversarial Networks (GANs) are supervised generative models that produce realistic and highly detailed data. This method involves training two neural networks: a generator, which generates fake data points, and a discriminator, which distinguishes between fake and real data points. The goal is to train the generator to generate data that the discriminator accepts as real. GANs excel in synthesizing images, videos, and unstructured data but require specialized knowledge for construction and training. They may also encounter issues where they produce a limited set of very similar fake data points.

Neural Radiance Fields (NeRFs) generate new views of a partially-known 3D scene. By interpolating a set of input images,

* Corresponding author

the algorithm adds new perspectives to the same object. A fully connected neural network treats the static scene as a continuous 5-dimensional function and predicts the content of each voxel. This technique is useful for generating realistic images from an existing set but suffers from slow training, slow rendering, and potential image quality or aliasing issues. Recent advancements in neural rendering algorithms aim to address these challenges.
Simulated Data involves using a virtual camera to generate physics-based and photorealistic simulations. This method includes all necessary annotations, dimensions, and labels to produce realistic 3D data. Simulated data offer flexibility in generating a wide range of scenarios and are suitable for complex scenarios. They allow adjustments to light conditions, texture modifications, colour variations, layout changes, object placement, and capturing rare real-world cases.
Each method has its advantages and limitations, and the choice depends on the specific application and requirements. For further insights into synthetic data, more detailed information can be found in specific literature reviews such as those conducted by Baraheem et al. (2023) and Man & Chahl (2022).

In the field of Architecture Engineering and Construction (AEC), several studies have suggested the utilization of synthetic data to enhance machine learning and deep learning algorithms for diverse applications. These works have developed various workflows for generating the required data. In this work (Hong et al., 2021), the authors proposed a three-step workflow for synthetic data generation for infrastructure scene understanding using building information models. The first step is to train a GAN network to enable the translation between photographs and BIM images. The second step is to generate labelled synthetic images that closely resemble photographs from the BIM images using the trained GAN, and the final step is to combine the synthetic images together in order to create a comprehensive dataset of high-quality synthetic data. Ma et al. (2020) conducted an investigation on the utilization of synthetic point cloud data for training deep models and facilitating the development of as-built BIM. They introduced a workflow that involved converting existing BIM models into synthetic point clouds using three different software tools. Subsequently, the generated data was employed to train a semantic segmentation model. Some researchers have proposed the generation of synthetic data for object recognition in construction site applications. The synthetic images are created by combining three-dimensional (3D) models of construction machines with various background images captured from construction sites (Soltani et al., 2016), or by utilizing rendered backgrounds (Barrera-Animas & Davila Delgado, 2023). In the heritage field few works dealing with synthetic data are available. In their research work (Tomalini et al., 2021), a methodology was proposed to enhance the training dataset required for developing software capable of recognizing architectural heritage using pictures captured from a mobile device. The proposed approach leverages Physically Based Rendering (PBR) tools. They devised a workflow that semi-automatically generates multiple rendered images from a 3D textured mesh. This involves defining camera positions around the building through a series of paths and rendering the scenes accordingly. To support CH point cloud classification, the authors in (Pierdicca et al., 2019) presented a novel framework for automatically generating synthetic point cloud datasets. The framework utilizes Blender, an open-source software that provides access to individual points in an object, allowing for the creation of new meshes. The described algorithms enable the generation of a large number of synthetic point clouds, simulating a virtual laser scanner at varying distances. Additionally, these algorithms can simultaneously generate multiple point clouds from a scene in Blender, including the use of existing models of ancient architectures. A first assessment of the use of such synthetic data was provided in (Morbidoni et al., 2020), in which the authors test a Dynamic Graph CNN trained for the semantic segmentation of CH. In this paper (Dulecha et al., n.d.), the authors presented SynthPS, a benchmark that encompasses synthetic, physically-based renderings of Cultural Heritage object models with various assigned materials. SynthPS allows assessing the performance of classical, robust, and learning-based Photometric Stereo approaches on materials with diverse light distributions. The study conducted by Garozzo et al. (2021) examined an approach that employed Generative Adversarial Networks (GANs) to automatically synthesize unrealistically composed photos. The aim was to overcome the scarcity of images available for training AI systems in the context of Cultural Heritage data understanding. The authors specifically proposed a method that utilized GAN techniques anchored to semantic ontology domain representation to guide the generation of realistic classical order images.

3. METHODOLOGY

The proposed methodology enables the creation of fully synthetic, simulated data by generating a series of rendered images from a detailed 3D model. While this study focuses on generating heritage building images, the method has the potential to be applied to various scenarios, allowing for the generation of images for objects of any type. The method consists of five primary steps: (i) generating the 3D model or scene of the object, (ii) applying and configuring the materials, (iii) setting and defining the object's views, (iv) configuring the scene with lighting and background, and (v) generating the rendered images along with their corresponding ground truth.
In the subsequent paragraph, we will develop into each phase in detail, using the case study of Spedale del Ceppo in Pistoia for illustration.

3.1 3D model generation

The first step in the procedure is to create 3D scenes or objects that simulate the target scenario. Depending on the required dataset, the 3D model can represent a real-world scene or object, or it can be partially or entirely created from scratch. In this study, we present a real-world case study, the Spedale del Ceppo, a Renaissance-era hospital in Pistoia built around the 13th century (Figure 1).

Figure 1. Photogrammetric point cloud of Spedale del Ceppo, Pistoia, Italy.

The building is part of the image-point dataset developed in (Pellis et al., 2021), which includes several data sources for each building such as the laser point cloud, photogrammetric images, and related photogrammetric point cloud. The 3D model of the building was manually created using the photogrammetric point cloud as a reference and developed with Autodesk Revit, resulting in a Building Information Model (BIM) composed of parametric elements. There are several advantages to using a

BIM model over a standard CAD model: (i) the creation of the 3D model is faster since standard parametric libraries of elements can be easily fitted to the real 3D elements, (ii) the model is quickly editable and customizable, allowing users to modify and change the main constructive elements of the building with several element typologies. Hence, it is possible to generate different scenes with the same model, increasing the variation in the final rendered images; (iii) The various element labels typical of a BIM model, such as material, family, instance, etc., allow for easier exporting of the 3D model, and thus, easier management during the scene setting and annotation phase. Figure 2 shows the BIM model of the Spedale del Ceppo.

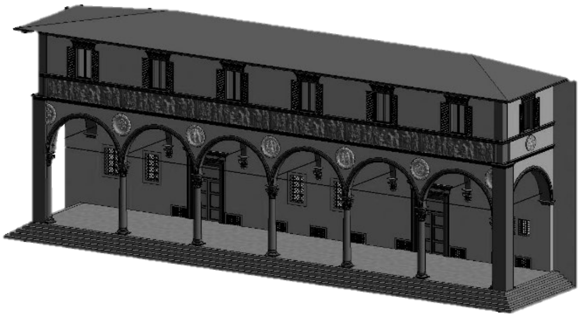

Figure 2. Building information model (BIM) of Spedale del Ceppo.

3.2 Material definition and application

The second step of the procedure involves the creation and the definition of the various materials and their application on the object surfaces. Material setting and application play a crucial role in the rendering process, significantly impacting the final visual quality and realism of a computer-generated image or animation. They determine how light interacts with surfaces, including properties such as colour, texture, reflectivity, transparency, and roughness. To create a proper functioning synthetic dataset with a good level of diversity and variance within the images, various material combinations can be applied to objects or the scene itself, allowing the model to predict a wide range of different objects. In the case study presented here, we aimed to simulate a real-world scenario and we applied a set of materials to the 3D model that were compatible with the existing building. There are various tools or software that allows the user to create, modify and apply materials to 3D object, and each software has its own unique interface and workflow for creating and applying material. Some popular licensed software are Autodesk 3ds Max, Autodesk Maya, Unreal Engine, or an open-source solution is Blender. Despite also Autodesk Revit has its tools for rendering, in this study we used the support of V-Ray, a powerful commercial rendering engine, that provides a variety of tools for fine-tuning the rendering process, including control over material. V-Ray has been used exploiting the Rhinoceros software, hence the 3D model has been exported from the BIM working space to the Rhinoceros environment. Some standard used materials were already available in the V-Ray library, such as glass, wood or plaster, and they have been set in colour, reflection or refraction, according with the surface to model. Other materials have been created from scratch with the Material Editor using textures or images, to simulate the building as much as possible. To enhance the visual realism of rendered objects, displacement maps and bump maps have been used. They allow to add details and surface irregularities to the 3D scene during the rendering, without altering the geometry of the object itself, creating the illusion of depth and texture. Figure 3 shows the 3D model after the application of the materials.

Figure 3. Render 3D model with materials.

3.3 View setting

The third step of the workflow involves setting up the views and the scene. Initially, to generate multiple rendered images of the 3D building model, various views of the model must be established. This can be achieved by positioning the camera at different locations and angles around the building. The camera can be rotated and tilted to capture diverse perspectives of the building. Furthermore, adjusting the camera's field of view allows control over the portion of the building visible in the image. Depending on the scene and the desired number of views, different camera positioning strategies can be employed. In this study, we propose designing a series of paths around the object or building to determine the camera points of view, along with specifying the number of views for each path. These paths can be linear, curved, or follow a complex trajectory depending on the requirements of the object. For each point, a camera is positioned within the scene, and several parameters need then to be set. These parameters include the camera's orientation, to determine its pointing direction and rotation within the scene, and the field of view, to control the amount of the scene captured by the camera. Moreover, depending on the rendering software being used, additional parameters can be set to fine-tune the camera's behaviour, such as aperture, shutter speed, ISO, depth of field, or other camera effects. Define camera paths offers several advantages: (i) it enables to have more flexibility to refine and to adjust the views, (ii) it can save time and allows to automatically generate multiple views and to automatize the rendering, (iii) it allows to obtain a comprehensive view of the object and its details, and (iv) it gives the possibility to create animation or video sequences. Figure 4 shows the paths that have been set around the study case building.

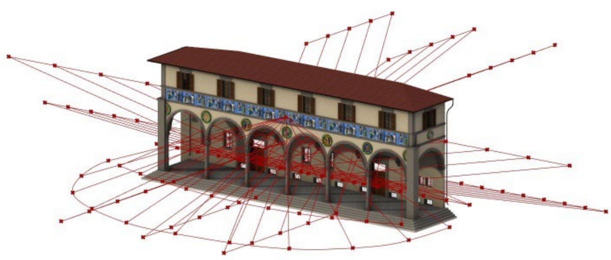

Figure 4. Render 3D model with materials.

They consist in six different paths around the building, and for each paths ten views have been set up. Hence, at the end of the procedure 60 different images are generated. The z-coordinate of the paths were positioned around 1.8 m to simulate a terrestrial photogrammetric acquisition.

3.4 Scene and light setting

In this phase the scene around the object needs to be set up. It involves the creation of the background and the light setting. Setting up the background for rendering involves creating an environment or backdrop that complements your scene and enhances its visual appeal. Depending on the application of the final rendered images it can be a solid colour, a gradient, an image, or a 3D environment. In the image generation presented in this work, we exploit a simplified 3D environment, simulating the background buildings with elementary 3D shapes, and we used the skydome to simulate the background sky . The skydome technique works by wrapping a spherical or hemispherical geometry with a texture or image representing the sky. This texture typically contains information about the sky appearance, including clouds, atmospheric effects, and the sun position. By using a skydome, the rendered scene receives ambient lighting and reflections that simulate the light coming from the sky, which contributes to the scene's overall lighting and realism. This technique is especially useful in outdoor scenes where the sky plays a significant role in the visual aesthetics and lighting conditions. Light setting is a fundamental phase in rendered data generation since it can greatly enhance the visual quality and realism of the 3D scene. It involves the definition of all the light in the scene, including their source types (point, directional, spotlight, area light) their intensity, positioning, colour, etc. Depending on the scene, the lights in the scene can be artificial, such as reflectors or diffusers, natural, such as the light of the sun, or a combination of both.
Since in this work we aim to simulate a real-world outdoor environment, we exploit environmental lights, without the use of artificial sources.

3.5 Image generation

The final phase involves generating the rendered images and the corresponding ground truth map. There are several render engines available, and we propose using V-Ray for Rhinoceros in our workflow. Multiple rendering options are available, which allow for improvements in both image quality and rendering times. First, the output size needs to be determined. In this test, we generated images with a size of 519 x 775 pixels, striking a balance between quality and rendering time. Secondly, the output quality can be adjusted using various parameters, including the noise limit, bucket size, shading rate, and min/max subdivisions. However, the availability of these options may vary depending on the render engine used, so further details can be found in the respective software manual. Figure 5 displays some of the generated images of the Spedale del Ceppo.

Figure 5. Example of rendered images.

After generation, the final number of images is 60. In order to increase the size of the dataset and enhance its variability, we generated a series of images with different skydomes and various sun positions, as shown in Figure 6. More specifically, we created 8 combinations, resulting in a total of 480 images.

Figure 6. Examples of rendered images with different light conditions and setting.

In addition to generating RGB images, it is necessary to create the corresponding ground truth maps. In this study, we propose generating a dataset that is compatible with the one described in (Pellis et al., 2021), following the guidelines outlined in the ARCHdataset (Matrone et al., 2020). To annotate the dataset, custom annotations must be directly set on the 3D model, and they are automatically generated during the rendering process. However, this procedure allows for the generation of various types of predefined output maps (as shown in Figure 7). These maps include the depth map, surface normals, object bounding boxes, materials, and many others.

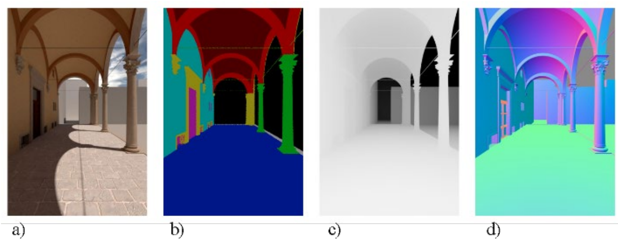

Figure 7. Example of semantic maps: a) RGB image, b) ARCH labels, c) depth map, d) surface normals.

4. TRAINING TESTS

To evaluate the proper functioning of the procedure, the quality of the generated images, and to assess the effect of synthetic data during training we performed a series of tests, using three state-of-the-art neural networks for semantic segmentation: Fully Convolutional Network (FCN), SegNet, and Deeplabv3+. The tests have been performed on the study case of Spedale del Ceppo. For this building, the following data resources were available: the BIM model, used to generate the synthetic images with the proposed procedure, and a set of real images of the building, acquired during the photogrammetric survey of the building, belonging to the heritage dataset developed in (Pellis et al., 2021). The real images have been labelled according with the method developed in the work of Pellis et al., 2021, and the same semantic maps were generated for the rendered images. Based on the available data, two main tests were performed.
Test 1. In the first test we evaluated the use of only synthetic data. For this purpose, all the synthetic generated images were combined to evaluate the model, by randomly shuffle them and splitting them in training (60%), validation (20%) and test set (20%). This test is similar to the test proposed in (Pellis et al., 2021) with the real images of Spedale del Ceppo.

Test 2. The second test evaluated whether the use of only synthetic images can generalize the real scenario, based on training and validating the model only on synthetic images, and testing the network only on real images.

4.1 Neural network architectures

In this work three neural networks for image segmentation have been used, and they are described in the following sections.

FCN (Long et al., 2014) is composed by a down-sampling part and an up-sampling part. The first part is a standard CNN composed by series of layers, in which the image features are extracted via convolution, followed by activation functions and pooling layers. At the end of the down-sampling network the number of channels is transformed into number of classes with a 1×1 convolutional layer. The up-sampling network transforms the height and width of the feature maps to those of the input image via *deconvolution* or *transposed convolution.*

Deeplabv3+ (Chen et al., 2018) uses atrous convolution with up-sampled filters to extract dense feature maps and capture long-range context. Atrous convolution enables explicit control over the density of feature computation and prevents signal decimation caused by stride and pooling. The encoder module encodes multi-scale contextual information by applying atrous convolution at multiple scales, while the simple but effective decoder module refines segmentation results along object boundaries.

SegNet (Badrinarayanan et al., 2017) is composed of an encoder network, a corresponding decoder network, and a final pixel-wise classification layer. The encoder network comprises 13 convolutional layers, and each encoder performs a convolution to generate a set of feature maps. The maps are then batch normalized and passed through an element-wise rectified linear unit (ReLU), which applies the function max(0,x). The decoding technique in SegNet involves convolving the feature maps with a trainable decoder filter bank to generate a dense feature map, which is then batch normalized. The final high-dimensional feature output from the last decoder is passed to a trainable softmax classifier.

4.2 Evaluation metrics

In order to evaluate the performance of our models, we utilized two evaluation metrics: Global Accuracy (GA) and mean Intersection Over Union (mIoU), which are defined by the equations below:

$$GA = \frac{\sum_i n_{ii}}{\sum_i t_i} \tag{4}$$

$$mIoU = \frac{1}{n_{cl}} \sum_i \frac{n_{ii}}{(t_i + \sum_j n_{ji} - n_{ji})} \tag{5}$$

where n_{cl} = number of classes included in ground truth
n_{ij} = number of pixels of class i predicted to belong class j
t_i = total number of pixels of class i in ground truth

In addition, we will display the confusion matrix for each model to provide a more in-depth analysis of the semantic segmentation performance.

4.3 Results

In this section, we present and compare the performances of the various models. First, we show and discuss the results obtained on Test 1. A comparison of the predicted maps with the three networks is showed, together with the obtained evaluation metrics. Second, we provide a detailed and extensive discussion of the results on Test 2, which are more representative of the usability of synthetic images in real-world scenarios.

Test 1

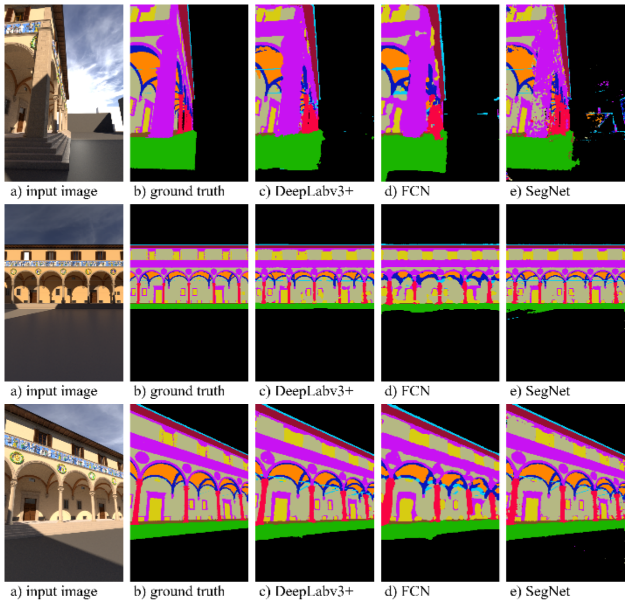

Figure 8. Test 1 – Prediction maps comparison between the three neural network models.

	GA	mean IoU	Mean F1
Deeplabv3+	0,97	0,84	0,94
FCN	0,90	0,74	0,82
SegNet	0,95	0,83	0,87

Table 1. Test 1 - Evaluation metrics comparison between the three models.

The initial tests yielded an impressive performance with all three models accurately predicting the output segmentation maps (Figure 8). According to Table 1, Deeplabv3+ outperformed the other models, and it yielded a GA of 97% and a mIoU of 84%. Additionally, the models trained with synthetic images showed better results compared to the models trained on real-world images of the building, as demonstrated in the test proposed by (Pellis et al., 2022b). This indicates the superior quality of the synthetic images, and the increased accuracy of the generated ground truth maps. However, the results are not remarkable in the real-world applications, and they are not representative of the capability of the models to predict unseen scenes.

Test 2

Fully Convolutional Network (FCN)

a) input image b) ground truth c) prediction

Figure 9. Test 2 – Prediction maps comparison with FCN: a) input image, b) ground truth, and c) prediction.

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	44.4	2.7	7.6	0.4	1.4	7.1	1.6	11.3	0.0	8.6	15.1
column	2.1	40.0	16.3	1.8	0.8	0.7	8.4	0.2	0.0	0.8	28.9
moldings	5.8	7.2	47.5	2.8	1.4	6.5	4.6	1.0	0.3	2.9	19.9
floor	0.3	0.7	2.5	17.0	0.7	0.5	36.0	0.0	0.0	0.1	42.2
door	4.1	10.4	34.7	6.7	7.7	4.4	3.4	1.6	0.2	2.0	24.9
wall	3.9	3.3	7.8	1.9	0.5	22.9	5.2	4.4	0.1	1.7	48.4
stair	0.0	3.6	4.2	10.9	0.4	0.8	60.7	0.0	0.0	0.2	19.1
vault	3.5	0.1	1.2	0.1	0.2	6.3	1.6	49.9	0.0	2.7	34.2
roof	7.0	6.4	30.3	0.6	5.3	3.8	0.7	2.4	32.8	6.7	3.9
other	7.1	4.6	17.0	1.7	2.9	2.9	2.8	3.5	1.3	43.0	13.2
none	1.2	5.0	9.0	9.8	1.2	3.8	11.7	0.1	0.3	1.4	56.3

Figure 10. Test 2 – Confusion matrix with Deeplabv3+.

	GA	mean IoU	Mean F1
FCN	0,39	0,21	0,32

Table 2. Test 2 - Evaluation metrics with FCN

SegNet

a) input image b) ground truth c) prediction

Figure 11. Test 2 – Prediction maps comparison with SegNet: a) input image, b) ground truth, and c) prediction.

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	61.6	1.5	6.3	0.0	1.5	7.5	0.0	8.8	0.2	5.1	7.4
column	2.1	49.8	11.3	0.6	0.7	1.5	0.7	0.1	0.1	2.6	30.5
moldings	3.9	7.3	48.9	2.8	2.9	4.9	0.8	0.2	0.2	3.7	24.4
floor	0.7	3.6	1.8	46.3	0.7	0.2	29.1	0.0	0.1	0.6	16.8
door	2.8	5.9	29.7	1.6	23.0	2.0	1.3	0.0	1.8	5.4	26.4
wall	3.2	2.5	5.9	0.3	1.5	24.5	0.3	0.4	0.2	1.3	59.8
stair	0.5	10.8	4.2	35.7	1.0	0.3	36.2	0.0	0.1	1.0	10.2
vault	4.3	0.1	0.8	0.0	0.2	29.7	0.0	47.6	0.0	0.8	16.3
roof	20.1	6.6	15.1	0.0	9.8	0.8	0.0	0.1	37.7	5.9	3.8
other	5.6	3.2	21.5	0.1	3.5	4.7	0.2	4.1	0.9	46.5	9.7
none	1.6	3.3	10.3	18.8	2.8	3.0	6.5	0.0	0.3	2.5	51.0

Figure 12. Test 2 – Confusion matrix with Deeplabv3+.

	GA	mean IoU	Mean F1
SegNet	0,43	0,26	0,36

Table 3. Test 2 - Evaluation metrics with SegNet

Deeplabv3+

a) input image b) ground truth c) prediction

Figure 13. Test 2 – Prediction maps comparison with Deeplabv3+: a) input image, b) ground truth, and c) prediction.

	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	60.8	0.3	5.8	0.1	0.6	7.7	0.2	7.6	0.0	6.0	10.8
column	0.5	53.7	21.2	2.0	2.3	3.5	0.5	0.1	0.0	0.7	15.4
moldings	1.1	1.4	69.8	3.8	2.8	6.4	0.5	0.2	0.3	1.0	12.9
floor	0.0	0.7	1.0	66.8	1.7	0.1	16.4	0.0	0.0	0.0	13.2
door	0.3	0.6	25.0	6.6	45.6	6.0	0.6	0.0	0.3	1.5	13.3
wall	1.1	0.8	11.0	3.4	1.7	40.5	0.2	0.6	0.2	0.7	39.9
stair	0.0	2.2	1.8	32.7	1.9	0.1	57.1	0.0	0.1	0.1	4.0
vault	2.1	0.0	0.2	0.0	0.0	15.6	0.1	58.0	0.0	1.3	22.7
roof	0.0	0.2	17.3	0.0	4.3	0.1	0.0	0.0	73.6	3.8	0.6
other	2.8	0.8	13.9	1.8	3.9	2.8	0.3	3.0	3.1	59.1	8.6
none	0.4	0.9	11.6	24.0	4.8	5.4	2.4	0.0	0.3	1.4	48.9

Figure 14. Test 2 – Confusion matrix with Deeplabv3+.

	GA	mean IoU	Mean F1
Deeplabv3+	0,54	0,40	0,48

Table 4. Test 2 - Evaluation metrics with Deeplabv3+.

4.4 Result discussion

This paragraph discusses the results of the conducted tests, highlighting the performance of different architectures in semantic segmentation. While the accuracies achieved in Test 1 were remarkable and consistent across all architectures, Deeplabv3+ emerged as the clear frontrunner in Test 2, surpassing the other two networks with a GA of 54% and an mIoU of 40%. FCN and SegNet demonstrated similar performance, both achieving a GA of approximately 40% and an mIoU of around 20%. Although the synthetic scenario was built based on the real building, and the real and the synthetic images showed several similarities, the network performances were not significantly noteworthy. However, upon analysing the image predictions and confusion matrices, certain observations can be made. Across all three architectures, the primary prediction errors stemmed from the network tendency to overpredict the "none" class. Rather than considering this an actual misclassification, it can be viewed as a failure to classify, attributable to the high occurrence of the "none" class in synthetic images. Addressing this issue could involve reducing the number of background pixels during image generation or applying weights to the class during training. This problem was particularly prominent with the "wall" class, which Deeplabv3+ predicted as "none" 40% of the time and SegNet predicted as "none" 60% of the time. The bright uniform texture of wall surfaces and overexposure in real images likely contributed to this misclassification. Generating synthetic images with diverse wall textures, lighting conditions, and exposure settings could mitigate this issue and improve the overall quality of synthetic images. Additionally, using correctly exposed images for testing could yield better results. Furthermore, the models displayed a tendency to accurately predict classes for background buildings, even when they were annotated as "none" in the ground truth. These predictions introduce bias and hinder the model overall performance, especially when evaluating correct predictions. However, this problem is less significant in applications where the background can be easily removed using other methods, such as when segmentation maps are used as an intermediate representation for 3D shape or point cloud segmentation (Pellis et al., 2022a).

5. CONCLUSION

This study presents a workflow for generating synthetic image data to facilitate the training and testing of machine learning systems for semantic segmentation. The process involves rendering multiple images from different perspectives and with varying lighting and scene conditions, all derived from a 3D model or scene. In the first part, the workflow was tested in the context of heritage building scenario generation, and specifically applied to the case study of Spedale del Ceppo. Results showed that the workflow is flexible and produces high-accuracy per-pixel segmentation maps, as well as various other annotation maps such as depth and surface normals. In the second part, a set of generated images were used to train and test three neural network architectures, and although the preliminary results were not exceptional, they indicate promise for future development and highlight the potential of synthetic data for large-scale image tasks. In addition, the procedure could be extended to other domains or scenarios beyond heritage buildings, such as medical imaging or robotics. Further improvements to the segmentation performance using synthetic data and their effect on training will be explored in future work. At first, refining the synthetic image generation workflow to produce even more realistic images and ground truth maps, and increasing the number and the diversity of training and testing images. Secondly, investigating the effectiveness of combining real and synthetic data in training and testing the neural networks. The focus of future developments will also include the production of new 3D models and scenarios to generate additional images, thereby increasing the variability within the synthetic dataset.

REFERENCES

Andrea Tomalini, Edoardo Pristeri, & Letizia Bergamasco. (2021). Photogrammetric Survey for a Fast Construction of Synthetic Dataset. In *Representation Challenges. Augmented Reality and Artificial Intelligence in Cultural Heritage and Innovative Design Domain*. FrancoAngeli srl. https://doi.org/10.3280/oa-686.34

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(12), 2481–2495. https://doi.org/10.1109/TPAMI.2016.2644615

Baraheem, S. S., Le, T. N., & Nguyen, T. V. (2023). Image synthesis: a review of methods, datasets, evaluation metrics, and future outlook. *Artificial Intelligence Review*. https://doi.org/10.1007/s10462-023-10434-2

Barrera-Animas, A. Y., & Davila Delgado, J. M. (2023). Generating real-world-like labelled synthetic datasets for construction site applications. *Automation in Construction*, *151*. https://doi.org/10.1016/j.autcon.2023.104850

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. (2018, February 7). Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. http://arxiv.org/abs/1802.02611

Dulecha, T. G., Pintus, R., Gobbetti, E., & Giachetti, A. (n.d.). *SynthPS: a benchmark for evaluation of Photometric Stereo algorithms for Cultural Heritage applications*. https://doi.org/10.2312/gch.20201288

Garozzo, R., Santagati, C., Spampinato, C., & Vecchio, G. (2021). Knowledge-based generative adversarial networks for scene understanding in Cultural Heritage. *Journal of Archaeological Science: Reports*, *35*. https://doi.org/10.1016/j.jasrep.2020.102736

Hong, Y., Park, S., Kim, H., & Kim, H. (2021). Synthetic data generation using building information models. *Automation in Construction*, *130*. https://doi.org/10.1016/j.autcon.2021.103871

Long, J., Shelhamer, E., & Darrell, T. (2014). *Fully Convolutional Networks for Semantic Segmentation*. http://arxiv.org/abs/1411.4038

Ma, J. W., Czerniawski, T., & Leite, F. (2020). Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds. *Automation in Construction*, *113*. https://doi.org/10.1016/j.autcon.2020.103144

Man, K., & Chahl, J. (2022). A Review of Synthetic Image Data and Its Use in Computer Vision. In *Journal of Imaging* (Vol. 8, Issue 11). MDPI. https://doi.org/10.3390/jimaging8110310

Matrone, F., Lingua, A., Pierdicca, R., Malinverni, E. S., Paolanti, M., Grilli, E., Remondino, F., Murtiyoso, A., & Landes, T. (2020). A Benchmark for Large-Scale Heritage Point Cloud Semanti Segmentation. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *43*(B2), 1419–1426. https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1419-2020

Morbidoni, C., Pierdicca, R., Paolanti, M., Quattrini, R., & Mammoli, R. (2020). Learning from synthetic point cloud data for historical buildings semantic segmentation. *Journal on Computing and Cultural Heritage*, *13*(4). https://doi.org/10.1145/3409262

Pellis, E., Masiero, A., Tucci, G., Betti, M., & Grussenmeyer, P. (2021). Assembling an Image and Point Cloud Dataset for Heritage Buildings Semantic Segmentation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVI-M-1–2021*, 539–546. https://doi.org/10.5194/isprs-archives-xlvi-m-1-2021-539-2021

Pellis, E., Murtiyoso, A., Masiero, A., Tucci, G., Betti, M., & Grussenmeyer, P. (2022a). 2D To 3D Label Propagation For The Semantic Segmentation Of Heritage Building Point Clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *43*(B2-2022), 861–867. https://doi.org/10.5194/isprs-archives-XLIII-B2-2022-861-2022

Pellis, E., Murtiyoso, A., Masiero, A., Tucci, G., Betti, M., & Grussenmeyer, P. (2022b). An Image-Based Deep Learning workflow for 3D Heritage Point Cloud Semantic Segmentation. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *XLVI-2/W1-2022*, 429–434. https://doi.org/10.5194/isprs-archives-XLVI-2-W1-2022-429-2022