A PERFORMANCE COMPARISON BETWEEN SEGNET AND DEEPLABV3+ ON THE SEMANTIC SEGMENTATION OF HERITAGE BUILDINGS

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

[1] Department of Civil and Environmental Engineering (DICEA), University of Florence, 50139 Florence, Italy - (eugenio.pellis, andrea.masiero, irene.cortesi, 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: image semantic segmentation, point cloud semantic segmentation, deep learning, heritage buildings

ABSTRACT:

During the last decade, the use of machine and deep learning tools to support 3D semantic segmentation of point clouds remarkably increased and their impressive results have led to the application of such methods to the semantic modeling of heritage buildings. Nevertheless, a standard procedure to deal with such problem is still missing, and several significant challenges, caused by the complexity of heritage building scenario, have still to be faced. This paper aims at comparing the overall performance of two convolutional neural network architectures, named SegNet and Deeplabv3+, for the semantic segmentation of heritage point clouds throughout a multiview approach. More specifically, the two architectures have been tested to obtain 2D segmentation maps of the related photogrammetric images of the buildings, and then the output maps have been projected to the photogrammetric point cloud by means of the interior and exterior camera parameters. Experiments to test the effectiveness of the proposed approach have been conducted on the case study of Spedale del Ceppo in Pistoia, Italy. Despite the results shown a remarkable performance of both the architectures, Deeplabv3+ outperformed SegNet in terms of accuracy, memory consumption and training time.

1. INTRODUCTION

Nowadays, deep learning methods are frequently used in many applications involving the need of smart data interpretation. In particular, they are used in a number of applications related to scene segmentation and understanding. In civil engineering, this can be useful for instance in order to support the creation of a building semantic model, e.g. a BIM (Building Information Modeling). Such operation is usually even more challenging in the heritage building case (Heritage Building Information Modeling, H-BIM), the case study considered in this paper, because of the peculiarities of heritage constructions.

Actually, different approaches have been considered in the literature for point cloud semantic segmentation. Taking into account of the already consolidated results obtained for image semantic segmentation, this work aims at the semantic segmentation of heritage building point clouds using a multi-view approach, similarly to (Pellis et al., 2022a).

More specifically, the point cloud segmentation is obtained in two steps:

- First, assuming that the considered point cloud is the outcome of a photogrammetric reconstruction, the images used to generate the point cloud are semantically segmented by using a deep learning-based network.
- Second, the image segmentation maps are transferred to the point cloud, by means of the interior and exterior camera parameters and a voting procedure.

The overall performance of such workflow is clearly highly dependent on the effectiveness of the first step. For this reason, this paper focuses on the performance comparison obtained when using different network architectures. To be more precise, the results obtained by using SegNet and DeepLabv3+ will be compared on a test building.

The proposed networks, which can be trained also on synthetic data in order to reduce the problem of collecting/generating a sufficient amount of data for the learning step (Man & Chahl, 2022), are trained to properly distinguish the classes defined in the ARCHdataset (Matrone, Lingua, et al., 2020). It is worth to notice that a similar approach can also be implemented considering different classes, and that the current choice has been motivated by the will of comparing the obtained results with others already obtained on similar problems and to ensure the possibility of integrating the considered dataset with pre-existing ones.

The paper is structured as follows: Section 2 is an overview of the related works dealing with semantic segmentation of heritage buildings; Section 3 introduces and explains the exploited network architectures; Section 4 illustrates the case study; Section 5 explains the training options and settings; Section 6 shows the results; Conclusion and future developments are discussed in Section 7.

2. RELATED WORKS

Several works have been proposed in the literature to address the problem of semantic segmentation of heritage buildings. Machine learning (ML) and deep learning (DL) are emerging in the architectural heritage domain as the preferred methods to support data interpretation and semantic enrichment of cultural heritage (Fiorucci et al., 2020), and in the last years they have been widely investigated for the automation of the semantic segmentation process. The authors in (Malinverni et al., 2019) proposed a method to label and automatically cluster a point

* Corresponding author

This contribution has been peer-reviewed.
https://doi.org/10.5194/isprs-archives-XLVIII-1-W1-2023-379-2023 | © Author(s) 2023. CC BY 4.0 License.

cloud based on a supervised deep learning approach, using the PointNet++ neural network. They underlined the bottlenecks of segmentation in the CH domain, mainly caused by complexity of the scenes. To address this limitation, they started to work on a specific fine-labelled dataset. In (Pierdicca et al., 2020), the authors propose a DL framework for Point Cloud segmentation, which employs an improved DGCNN (Dynamic Graph Convolutional Neural Network) by adding meaningful features such as normals and colours. The approach has been applied to the ARCHdataset, a dataset specifically built for training and testing learning segmentation approaches. The experiments achieved high accuracy, demonstrating the effectiveness and suitability of the proposed approach compared to other methods. (Matrone, Grilli, et al., 2020) made a comparison between machine and deep learning methods for large 3D cultural heritage classification. Then, considering the best performances of both techniques, they proposed an architecture named DGCNN-Mod+3Dfeat that combines the positive aspects and advantages of these two methodologies for semantic segmentation of cultural heritage point clouds. (Murtiyoso & Grussenmeyer, 2020) proposed an automated pipeline for segmenting and classifying multi-scalar point clouds in the context of heritage objects. Multi-level segmentation is performed from historical neighbourhood scale up to the scale of architectural elements, such as pillars and beams. They proposed an algorithmic approach in the form of a toolbox, which includes several functions for semantically segmenting large point clouds into smaller, more manageable, and semantically labelled clusters. (Cao et al., 2022) presented and compared two different approaches for the 3D semantic segmentation task in the heritage field, e.g. on point clouds of three chapels of the "Sacromonte Calvario di Domodossola" and two scenes from the ArCH dataset. The authors used a ML method based on the Random Forest (RF) classifier. Then, they employed dynamic graph convolutional neural network (DGCNN) as DL method, training on the ArCH dataset and testing on both the two unseen test scenes of the ArCH dataset and on the “Sacrimonti” chapel point clouds. According to their comparison of DL-based and ML-based methods, the DL method is less generalizable, but it extracts features and test scenes more efficiently without the need for manual labeling during classification. ML, on the other hand, requires specific training for each test case and manual segmentation of samples. In (Grilli et al., 2019), the authors provided a general method to classify heritage point clouds based on geometric covariance features. They analysed the impact of different features calculated on spherical neighbourhoods, varying the neighbourhood size, and they found the optimal radius. Achieved results indicate that to obtain correct classifications, it is not necessary to use a lot of features extracted at many different scales. Indeed, the adaptive size strategy allows the retrieval of better results in a shorter time. In (Croce et al., 2021) and (Croce et al., 2023), the authors exploited a supervised machine learning algorithm, that allows to propagate the manual annotation on a reduced portion of the point cloud to the whole point cloud via a Random Forest classifier by selecting a set of features in a chosen local neighbourhood of each 3D point. These features are either related to the mutual position of the points in the 3D space to their colour, or laser scanning intensity information.

3. NEURAL NETWORK MODELS

In this work two state-of-the-art neural networks for image semantic segmentation have been tested and compared: SegNet and Deeplabv3+. They have been implemented with MATLAB and they are described in the following sections. In both the cases the behaviour of the considered (pre-trained) networks have been optimized on the considered training dataset.

SegNet. SegNet (Badrinarayanan et al., 2017) is composed by an encoder network and a corresponding decoder network, followed by a final pixel-wise classification layer. The encoder network consists of 13 convolutional layers, and each encoder performs a convolution to produce a set of feature maps. The maps are then batch normalized and passed through an element-wise rectified linear unit (ReLU)max(0,x). Following that, a 2×2 window with stride 2 max-pooling layer is applied, and the result sub-sampled by a factor of 2. To avoid loss of spatial resolution it is necessary to capture and store boundary information before max-pooling and sub-sampling. The decoder network has the same number of layers of the encoder, and it up-samples the input feature maps using the memorized man-pooling indices. The SegNet decoding technique consists in convolving the feature maps with a trainable decoder filter bank to produce a dense feature map that is then batch normalized. The final high dimensional feature output, produced by the last decoder, is fed to a trainable soft-max classifier. The output is a K channel image of probabilities where K is the number of classes. In MATLAB the function segnetLayers() returns the SegNet architecture. It requires the specification of the input image size, the number of categories and the choice of a base model. The available models are VGG-16 and VGG-19, with an encoder depth of 5, pretrained on ImageNet database. The results presented in this study are carried out with VGG-19.

Deeplabv3+. Deeplabv3+ (Chen et al., 2018) employs atrous convolution with up-sampled filters to extract dense feature maps and to capture long range context. Atrous convolution allows to explicitly control how densely to compute the feature, and it allows to avoid 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 yet effective decoder module refines the segmentation results along object boundaries. MATLAB allows the implementation of this network architecture with the function deeplabv3plus(), that requires three inputs: the image size, specified as a 2-element or 3-elements vector in the format [height, width, 3], the number of the classes, specified as an integer greater than 1, and the classification network. Several architectures are available, with different characteristics, mainly differing on precision, speed, and network dimension. The choice of the architecture is based on a compromise between these characteristics. In this study, four based architectures have been tested: ResNet18, ResNet50, VGG-16, VGG-19. After several tests turned out that ResNet18 was the most suitable on the considered data, and the best compromise between speed and precision. The results that are going to be illustrated are the results obtained with ResNet18 (He et al., 2015), pretrained on the ImageNet database.

4. THE DATASET

The dataset developed in (Pellis et al., 2021) has been used to compare the two neural network models. The dataset is composed by five heritage buildings, and for each building three types of data are available: (i) the laser point cloud, (ii) the photogrammetric images and (iii) the related point clouds. All the data are labelled according to the guidelines of ARCHdataset, hence the images and the point clouds are annotated in 10 categories, corresponding to the main BIM standard elements. This dataset is particularly suitable for the development and the testing of a multiview-based semantic segmentation procedure. The comparison in this work has been performed using one building of the dataset, the Spedale del Ceppo in Pistoia, Italy. The two point clouds with the respective ground-truth, and some images with the respective ground-truth of the study case are illustrated in Figure 1 and Figure 2. The histogram in Figure 3 shows the balance of the classes of the study case for all the three data typologies, and it shows a remarkable class imbalance.

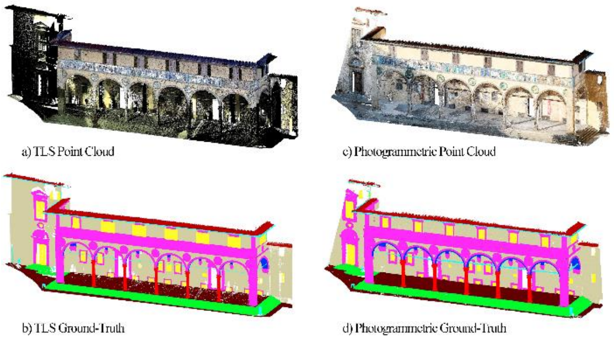

a) TLS Point Cloud
c) Photogrammetric Point Cloud
b) TLS Ground-Truth
d) Photogrammetric Ground-Truth

Figure 1. (1_SC) Spedale del Ceppo: a) TLS point cloud, b) TLS ground-truth, c) photogrammetric point cloud, d) photogrammetric ground-truth.

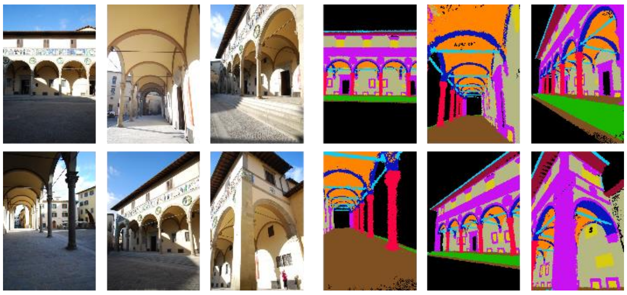


Figure 2. (1_SC) Spedale del Ceppo: a) RGB images and b) respective ground-truth.

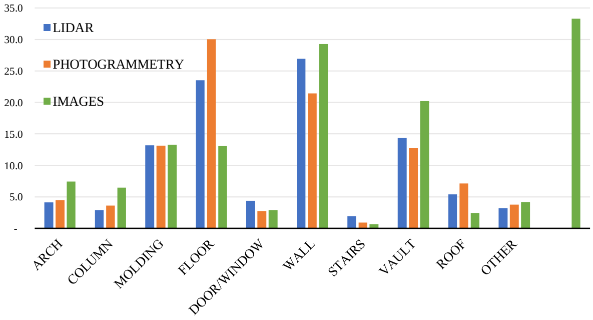


Figure 3. Class percentage distribution for the TLS clouds (blue), for the photogrammetric clouds (orange), and for the images (green).

5. TRAINING SETTINGS

In this section the settings used to evaluate the performance of the two models will be illustrated, and they include the image processing and preparation (§5.1), the training tests (§5.2), the hyperparameter tuning (§5.3), and the evaluation metrics (§5.4).

5.1 Image processing and preparation

For the tests, before starting the training procedure, the images generated by the labelling projection procedure have been processed and prepared to make them suitable to feed the network. Each image has been processed by means of a three-step workflow:

Resizing. To maintain the highest quality and accuracy, the ground-truth output of the labelling procedure has been produced with the same dimension of the input images, and, initially, the images of the case study had the dimension of 2592×3872 pixels. This input size is too large to train a deep network, and it would require long training time and high memory consumption. Hence, the images have been downsized to 720×1075 pixels. Furthermore, this operation allows to homogenize data of different size in case of an integration with new images captured with different cameras or sensors.

Verticality. Since the photogrammetric survey images could be acquired with different camera orientations, they may not always accurately depict the scene's verticality. The building or scene has been rotated to maintain the correct verticality on each image during the training so that the network can learn some features more easily.

Cropping. As a result of the rotation, the image could have different aspect ratios between width and height, but the neural network needs the same input size for training. For each image, two overlapping square tiles have been created to avoid resizing and distortion. Hence, the final size of the input is 720×720 pixels with 3 channels (RGB).

5.2 Tests

Data distribution and splitting are two key points to structure a machine learning test, as they have a remarkable effect on the model performance and usability. The test considered in this work have the aim to compare two network architectures on the semantic segmentation of a building. The entire set of images of the considered building was randomly shuffled, and then partitioned in training, validation and test set, with the percentage of 60%, 20%, and 20%, respectively. Since the images in the test set are similar to the images in the training set, the model should be able to generalize the solutions quite easily in this test. Despite this test does not provide a general model with a wide capability, it is helpful to easily compare the performance of different architectures, to assess the quality and the correct functioning of the developed segmentation procedure, and, generally, to conduct easy preliminary evaluations. Figure 4 shows the data splitting for the test using the study case building (1_SC) Spedale del Ceppo.

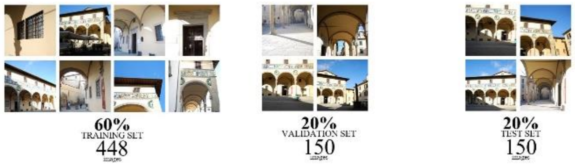


Figure 4. Training data structure and splitting for (1_SC).

5.3 Hyperparameter tuning

In deep learning tasks, hyperparameters control the optimization algorithm used in the learning phase. An appropriate choice of the hyperparameters is important for an efficient training convergence, and for an optimal performance achievement. Hyperparameter tuning consists of finding a set of optimal hyperparameters to be used during the learning phase. A summary of the most important hyperparameters is reported in the following.

Learning Rate. It is a hyperparameter that controls how much to change the model in response to the estimated error each time the model weights are updated. Too small learning rate may result in long training, while too large rate may result in an unstable training process. After a series of tests, the initial learning rate was set to α = 0.001 with a drop during training, updating the value every 5 epochs with a factor of 0.3.

Batch Size. It is the size of the mini-batch to use for each training iteration. A mini-batch is a subset of the training set that is used to evaluate the gradient of the loss function and update the weights. A large batch size allows a faster convergence but is more computationally expensive and lead to poor generalization. The size was set from 4 to 8, as a compromise between memory consumption and fast convergence.

Loss Function. It is the function that maps onto a numerical value the difference between the predicted label $\hat{y}$ and the ground truth label y_{GT} during the training. Various loss functions have been proposed in literature, and a detailed survey on existing loss function for semantic segmentation can be found in (Jadon, 2020). In the proposed tests the Cross-Entropy loss is used (Ma et al., 2004), and it is defined as a measure of the difference between two probability distribution for a given set of events. It is defined as follow:

$$L_{BCE}(y_{GT}, \hat{y}) = -(y_{GT}\log(\hat{y}) + (1 - y_{GT})\log(1 - \hat{y})) \tag{1}$$

Optimizer. The optimizer or solver is used to update the parameters at each iteration during training to minimize the loss function. There are many optimizers, and the choice among them is an important aspect to perform a good training. In this study three optimizers have been tested. The Stochastic Gradient Descent (SGD), the Root Mean Square Propagation (RMSProp) and the Adam. After a series of tests, the SGD with Momentum turned out to be the most suitable. The SGD algorithm updates the weight and biases to minimize the loss function, by determining small steps at each iteration in the direction of the negative gradient of the loss, but it can oscillate along the path towards the optimum. The Stochastic Gradient Descent with Momentum (SGDM) reduces this oscillations adding an additional term. It is defined as follows:

$$\theta_{\ell+1} = \theta_\ell - \alpha \nabla E(\theta_\ell) + \gamma(\theta_\ell - \theta_{\ell-1}) \tag{2}$$

where ℓ is the iteration number, α>0 is the learning rate, θ is the parameter vector, $E(\theta)$ is the loss function, and γ is the momentum. More detailed information about the optimizers can be found in (Choi et al., 2019).

L2 Regularization. In order to reduce the overfitting, an additional regularization term can be inserted in the loss function:

$$E_R(\theta) = E(\theta) + \lambda\Omega(w) \tag{3}$$

Where w is the weight vector, and λ is the regularization factor. After a series of tests the regularization factor was set to λ=0,005.

Class Weighting. As already shown previously, the classes of the case study are not balanced. To improve the performance when class imbalance is present, class weighting can be used. Class weights define the relative importance of each class in the training process. They can be set inversely proportional to the frequency of the respective classes, therefore increasing the importance of less prevalent classes.

N° of Epochs. It is the maximum number of epochs during the training. One epoch corresponds to the completion of a forward and backward passage through the neural network of the entire dataset. As the number of epochs increases, the weights are changed a greater number of times in the neural network. With the increase of the number of the epochs, the obtained results typically change from underfitting, to optimal, to overfitting. Experiments have shown that over 30 epochs there was no remarkable benefits in terms of loss, hence the maximum was set to 35 epochs.

5.4 Evaluation metrics

To evaluate the performance of our models we used two evaluation metrics: the Global Accuracy (GA), and the mean Intersection Over Union (mIoU) defined in 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

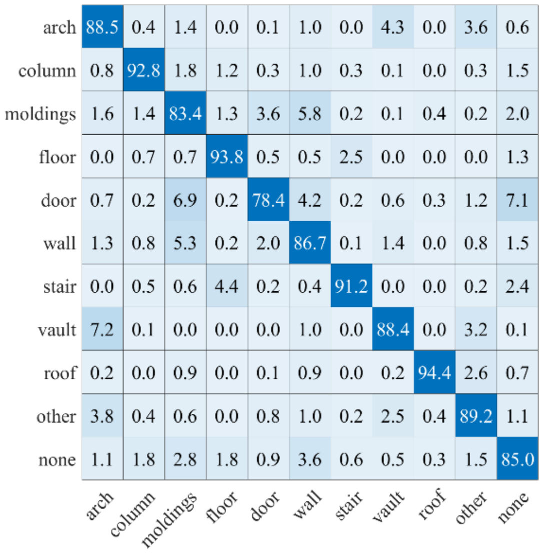
For each model, the confusion matrix will be shown as well, in order to provide a more in-depth analysis of the semantic segmentation performance.
In the next sections we are going to show at first the results for the image segmentation (6.1), and secondly the final results on the point clouds (6.2).

6. RESULTS

In this section the results are illustrated. At first, the results on image semantic segmentation are shown, both with SegNet and Deeplabv3+. Secondly, the results on point cloud segmentation are illustrated. They are the outcomes of the labelling projection procedure introduced in (Pellis et al., 2022b). For each test the GA, the mIoU and the confusion matrices and shown, together with some predicted segmentation maps.

6.1 Image segmentation

SegNet



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	88.5	0.4	1.4	0.0	0.1	1.0	0.0	4.3	0.0	3.6	0.6
column	0.8	92.8	1.8	1.2	0.3	1.0	0.3	0.1	0.0	0.3	1.5
moldings	1.6	1.4	83.4	1.3	3.6	5.8	0.2	0.1	0.4	0.2	2.0
floor	0.0	0.7	0.7	93.8	0.5	0.5	2.5	0.0	0.0	0.0	1.3
door	0.7	0.2	6.9	0.2	78.4	4.2	0.2	0.6	0.3	1.2	7.1
wall	1.3	0.8	5.3	0.2	2.0	86.7	0.1	1.4	0.0	0.8	1.5
stair	0.0	0.5	0.6	4.4	0.2	0.4	91.2	0.0	0.0	0.2	2.4
vault	7.2	0.1	0.0	0.0	0.0	1.0	0.0	88.4	0.0	3.2	0.1
roof	0.2	0.0	0.9	0.0	0.1	0.9	0.0	0.2	94.4	2.6	0.7
other	3.8	0.4	0.6	0.0	0.8	1.0	0.2	2.5	0.4	89.2	1.1
none	1.1	1.8	2.8	1.8	0.9	3.6	0.6	0.5	0.3	1.5	85.0

Figure 5. SegNet results on images: confusion matrix.

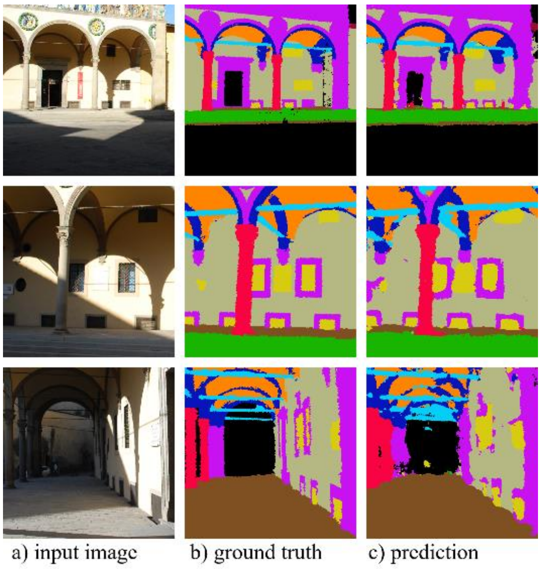


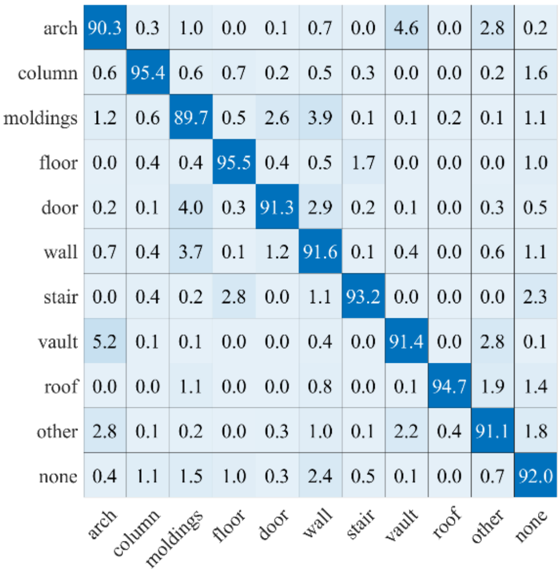
Figure 6. Prediction comparison with SegNet: a) input images, b) ground-truth, c) predictions.

	GA	mean IoU	Mean F1
SegNet	0,87	0,72	0,67

Table 1. Evaluation metrics on images for SegNet.

The results obtained with SegNet are satisfactory, and the model yielded a GA of 87% and a mIoU of 72% (Table 1). Generally, all the classes are well predicted, with no remarkable errors. The confusion matrix (Figure 5) shows that the errors are mainly focused on: (i) the class "arch" often confused with the class "vault" and vice versa; (ii) the class "moulding" sometimes confused with "door/window" or "wall", and (iii) the class "door/window" predicted such as "moulding" or "none". The class "none" is generally confused with all the classes because of its variable content.

DeepLabv3+



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	90.3	0.3	1.0	0.0	0.1	0.7	0.0	4.6	0.0	2.8	0.2
column	0.6	95.4	0.6	0.7	0.2	0.5	0.3	0.0	0.0	0.2	1.6
moldings	1.2	0.6	89.7	0.5	2.6	3.9	0.1	0.1	0.2	0.1	1.1
floor	0.0	0.4	0.4	95.5	0.4	0.5	1.7	0.0	0.0	0.0	1.0
door	0.2	0.1	4.0	0.3	91.3	2.9	0.2	0.1	0.0	0.3	0.5
wall	0.7	0.4	3.7	0.1	1.2	91.6	0.1	0.4	0.0	0.6	1.1
stair	0.0	0.4	0.2	2.8	0.0	1.1	93.2	0.0	0.0	0.0	2.3
vault	5.2	0.1	0.1	0.0	0.0	0.4	0.0	91.4	0.0	2.8	0.1
roof	0.0	0.0	1.1	0.0	0.0	0.8	0.0	0.1	94.7	1.9	1.4
other	2.8	0.1	0.2	0.0	0.3	1.0	0.1	2.2	0.4	91.1	1.8
none	0.4	1.1	1.5	1.0	0.3	2.4	0.5	0.1	0.0	0.7	92.0

Figure 7. Deeplabv3+ results on images: confusion matrix.

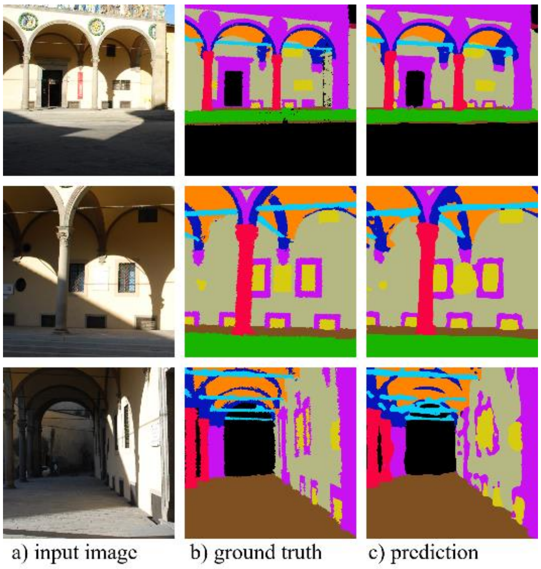


Figure 8. Prediction comparison with Deeplabv3+: a) input images, b) ground-truth, c) predictions.

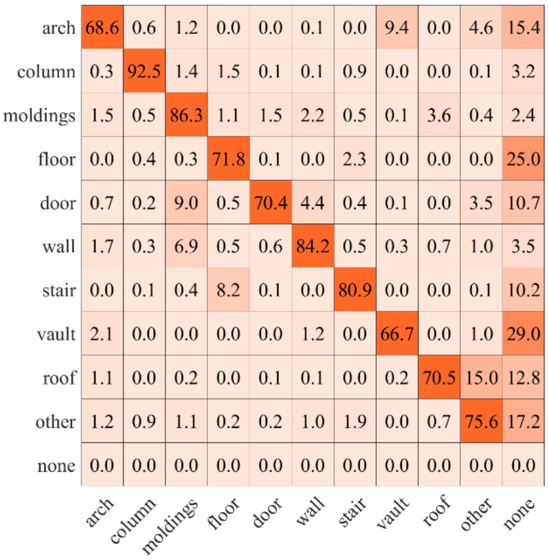
	GA	mean IoU	Mean F1
Deeplabv3+	0,92	0,81	0,80

Table 2. Evaluation metrics on images for Deeplabv3+.

The results obtained with Deeplabv3+ are satisfactory as well: the model yielded a GA of 92% and a mIoU of 81% (Table 2), overcoming the SegNet performance. The errors are mainly focused in the same categories, as shown by the confusion matrix (Figure 6). It is worth to notice that to reach a good convergence and a loss plateau, SegNet was trained for 60 epochs, with a medium time for each epoch of 8-10 min using a GPU GeForce RTX 4090 24 GB. In order to obtain a similar result, Deeplabv3+ was trained for 30 epochs, with a medium time for each epoch of 6-8 min.

6.2 Point cloud segmentation

SegNet



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	68.6	0.6	1.2	0.0	0.0	0.1	0.0	9.4	0.0	4.6	15.4
column	0.3	92.5	1.4	1.5	0.1	0.1	0.9	0.0	0.0	0.1	3.2
moldings	1.5	0.5	86.3	1.1	1.5	2.2	0.5	0.1	3.6	0.4	2.4
floor	0.0	0.4	0.3	71.8	0.1	0.0	2.3	0.0	0.0	0.0	25.0
door	0.7	0.2	9.0	0.5	70.4	4.4	0.4	0.1	0.0	3.5	10.7
wall	1.7	0.3	6.9	0.5	0.6	84.2	0.5	0.3	0.7	1.0	3.5
stair	0.0	0.1	0.4	8.2	0.1	0.0	80.9	0.0	0.0	0.1	10.2
vault	2.1	0.0	0.0	0.0	0.0	1.2	0.0	66.7	0.0	1.0	29.0
roof	1.1	0.0	0.2	0.0	0.1	0.1	0.0	0.2	70.5	15.0	12.8
other	1.2	0.9	1.1	0.2	0.2	1.0	1.9	0.0	0.7	75.6	17.2
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 9. SegNet projection results: confusion matrix.

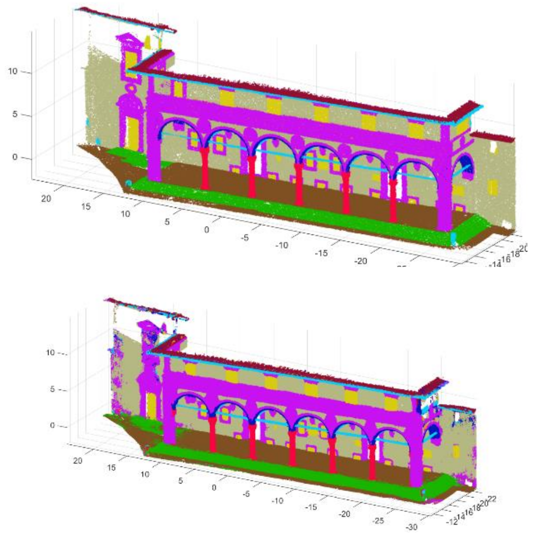
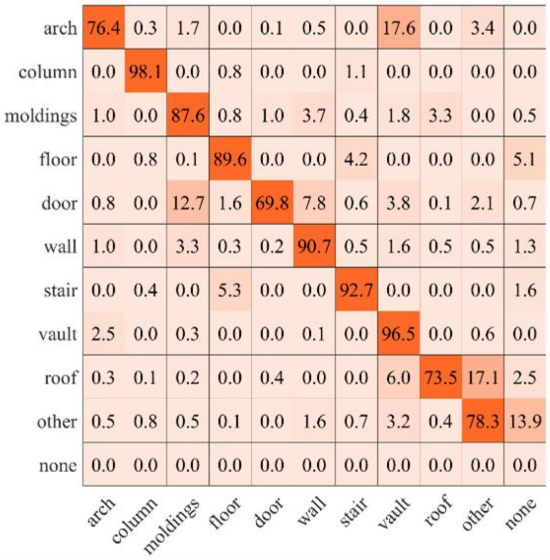


Figure 10. Point cloud prediction comparison with SegNet: a) ground-truth point cloud, b) predicted point cloud.

	GA	mean IoU	Mean F1
SegNet	0,85	0,66	0,63

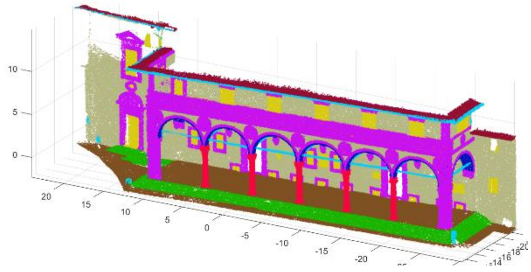
Table 4. Point cloud evaluation metrics for SegNet.

Deeplabv3+



	arch	column	moldings	floor	door	wall	stair	vault	roof	other	none
arch	76.4	0.3	1.7	0.0	0.1	0.5	0.0	17.6	0.0	3.4	0.0
column	0.0	98.1	0.0	0.8	0.0	0.0	1.1	0.0	0.0	0.0	0.0
moldings	1.0	0.0	87.6	0.8	1.0	3.7	0.4	1.8	3.3	0.0	0.5
floor	0.0	0.8	0.1	89.6	0.0	0.0	4.2	0.0	0.0	0.0	5.1
door	0.8	0.0	12.7	1.6	69.8	7.8	0.6	3.8	0.1	2.1	0.7
wall	1.0	0.0	3.3	0.3	0.2	90.7	0.5	1.6	0.5	0.5	1.3
stair	0.0	0.4	0.0	5.3	0.0	0.0	92.7	0.0	0.0	0.0	1.6
vault	2.5	0.0	0.3	0.0	0.0	0.1	0.0	96.5	0.0	0.6	0.0
roof	0.3	0.1	0.2	0.0	0.4	0.0	0.0	6.0	73.5	17.1	2.5
other	0.5	0.8	0.5	0.1	0.0	1.6	0.7	3.2	0.4	78.3	13.9
none	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

Figure 11. Deeplabv3+ projection results: confusion matrix.



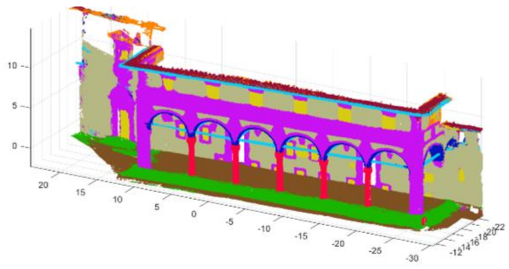


Figure 12. Point cloud prediction comparison with Deeplab: a) ground-truth point cloud, b) predicted point cloud.

	GA	mean IoU	Mean F1
Deeplabv3+	0,91	0,76	0,74

Table 5. Point cloud evaluation metrics for Deeplabv3+.

The results on point cloud segmentation showed a good performance of the labelling procedure, since the performance level obtained on image segmentation from both the networks has been quite maintained also on for the point cloud, without a remarkable degradation or information loss. The results are confirmed also graphically by looking at the point clouds, which turned out to be correctly annotated, with the exception of some small areas. The confusion matrices confirm that the errors are mainly focused on the class "arch", which is often confused with the class "vault", on the class "window/door", occasionally predicted as "moulding" or "wall", hence confirming the same trend already noted on image segmentation.
The histograms below report the final comparisons between the two neural networks, both for image (Figure 13) and point cloud (Figure 14) segmentation.

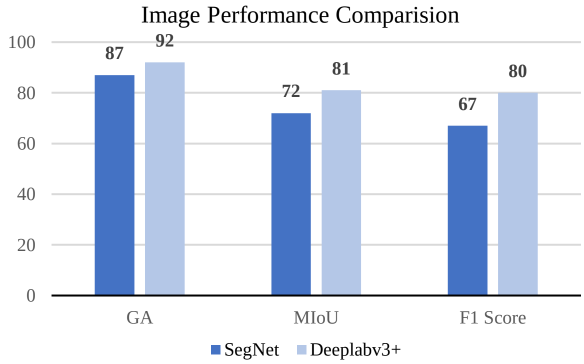


Figure 13. Performance comparison on image segmentation.

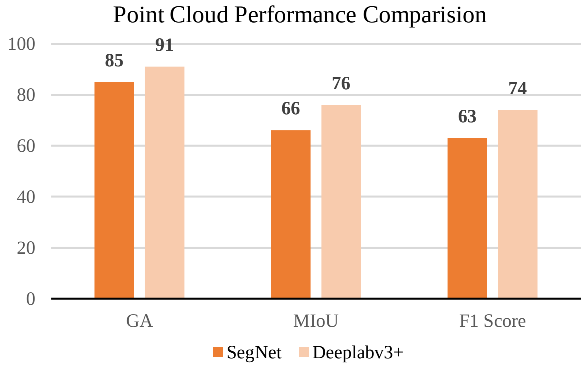


Figure 14. Performance comparison on point clouds segmentation.

Currently Deeplabv3+ turned out the most suitable to be integrated in the multiview point cloud segmentation workflow. It overcomes the SegNet performance on image segmentation, and, consequently, on point cloud segmentation as well. Moreover, Deeplabv3+ turned out to be more efficient in terms of GPU memory consumption and training time. Indeed, in order to yield the reported performance, SegNet required a 60 epochs training, with a total training time of 580 min. Instead, Deeplabv3+ required 30 epochs with a total training time of 250 min. Moreover, Deeplabv3+ achieved the 90% of the final performance after 20 epochs.

7. CONCLUSION

In this work, we showed a comparison between the performances of two state-of-the-art semantic segmentation architectures for the semantic segmentation of heritage building point clouds, throughout a multiview-based approach. Specifically, SegNet and Deeplabv3+ have been tested on the study case of Spedale del Ceppo in Pistoia, Italy. It is worth to notice that, despite both the considered nets allowed to obtain quite reasonable results, DeepLabv3+ quite clearly outperformed SegNet, while also ensuring a reduction of the training time. The results obtained for the point clouds confirm the trend: DeepLabv3+ appears to be a better choice for this kind of problem. Furthermore, transferring the semantic segmentation from images to point clouds does not have a significant impact on the results in the considered case, i.e. the semantic segmentation performance obtained for point clouds is quite similar to that in the image case. In order to improve the overall segmentation results, future developments will be done in particular to improve the image segmentation performance, including (i) the testing of other semantic or instance segmentation architectures, such as Mask-CNN, (ii) the integration of additional features during training, like the depth or the surface normal, (iii) the use of synthetic data to improve the generalization of the network, (iv) the integration of 3D point clouds to develop an image-point based approach. Future experiments will be performed to test and compare the procedure also on other study cases and also on unseen scenarios to test the capability of the network to predict new scenes.

REFERENCES

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

Cao, Y., Teruggi, S., Fassi, F., & Scaioni, M. (2022). A Comprehensive Understanding of Machine Learning and Deep Learning Methods for 3D Architectural Cultural Heritage Point Cloud Semantic Segmentation. *Communications in Computer and Information Science*, *1651 CCIS*, 329–341. https://doi.org/10.1007/978-3-031-17439-1_24

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

Choi, D., Shallue, C. J., Nado, Z., Lee, J., Maddison, C. J., & Dahl, G. E. (2019). *On Empirical Comparisons of Optimizers for Deep Learning*. http://arxiv.org/abs/1910.05446

Croce, V., Caroti, G., De Luca, L., Jacquot, K., Piemonte, A., & Véron, P. (2021). *From the Semantic Point Cloud to Heritage-Building Information Modeling: A Semiautomatic Approach Exploiting Machine Learning*. https://doi.org/10.3390/rs

Croce, V., Caroti, G., Piemonte, A., De Luca, L., & Véron, P. (2023). H-BIM and Artificial Intelligence: Classification of Architectural Heritage for Semi-Automatic Scan-to-BIM Reconstruction. *Sensors*, *23*(5). https://doi.org/10.3390/s23052497

Fiorucci, M., Khoroshiltseva, M., Pontil, M., Traviglia, A., Del Bue, A., & James, S. (2020). Machine Learning for Cultural Heritage: A Survey. *Pattern Recognition Letters*, *133*, 102–108. https://doi.org/10.1016/j.patrec.2020.02.017

Grilli, E., Farella, E. M., Torresani, A., & Remondino, F. (2019). Geometric Features Analysis for the Classification of Cultural Heritage Point Clouds. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2/W15), 541–548. https://doi.org/10.5194/isprs-archives-XLII-2-W15-541-2019

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *Proceedings of the IEEE Conference OnComputer Vision and Pattern Recognition*, 770–778. http://arxiv.org/abs/1512.03385

Jadon, S. (2020). *A survey of loss functions for semantic segmentation*. https://doi.org/10.1109/CIBCB48159.2020.9277638

Ma, Y. De, Liu, Q., & Qian, Z. B. (2004). Automated image segmentation using improved PCNN model based on cross-entropy. *2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, ISIMP 2004*, 743–746. https://doi.org/10.1109/isimp.2004.1434171

Malinverni, E. S., Pierdicca, R., Paolanti, M., Martini, M., Morbidoni, C., Matrone, F., & Lingua, A. (2019). Deep Learning for Semantic Segmentation of 3D Point Cluod. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2/W15), 735–742. https://doi.org/10.5194/isprs-archives-XLII-2-W15-735-2019

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., Grilli, E., Martini, M., Paolanti, M., Pierdicca, R., & Remondino, F. (2020). Comparing machine and deep learning methods for large 3D heritage semantic segmentation. *ISPRS International Journal of Geo-Information*, *9*(9). https://doi.org/10.3390/ijgi9090535

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

Murtiyoso, A., & Grussenmeyer, P. (2020). Virtual disassembling of historical edifices: Experiments and assessments of an automatic approach for classifying multi-scalar point clouds into architectural elements. *Sensors (Switzerland)*, *20*(8). https://doi.org/10.3390/s20082161

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). 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

Pellis, E., Murtiyoso, A., Masiero, A., Tucci, G., Betti, M., & Grussenmeyer, P. (2022b). 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

Pierdicca, R., Paolanti, M., Matrone, F., Martini, M., Morbidoni, C., Malinverni, E. S., Frontoni, E., & Lingua, A. M. (2020). Point cloud semantic segmentation using a deep learning framework for cultural heritage. *Remote Sensing*, *12*(6). https://doi.org/10.3390/rs12061005