SEMANTIC POINT CLOUD SEGMENTATION IN URBAN ENVIRONMENTS WITH 1D CONVOLUTIONAL NEURAL NETWORKS

S. M. González-Collazo, N. Canedo-González, E. González, J. Balado*

GeoTECH, CINTECX, Universidade de Vigo, 36310 Vigo, Spain
(silvgonzalez,elena,jbalado)@uvigo.gal, ncanedo@alumnado.uvigo.gal

KEY WORDS: Machine Learning, Deep Learning, Mobile Laser Scanning, urban environment, geometric features.

ABSTRACT:

Convolutional Neural Networks (CNNs) have been widely recognized for their efficacy in image analysis tasks. This paper investigates the application of the 1D-CNN variant CNNs for the semantic segmentation of urban point clouds obtained through Mobile Laser Scanning. Ten well-known local geometric features of point clouds were used as input for the 1D CNN. Through an empirical analysis on the Santiago Urban Dataset, the 1D CNN was optimized in terms of numbers of convolution layers, neurons, pooling layers, dropout layers, dense layers, training epochs, and batch size. The performance of the proposed 1D CNN was compared with Support Vector Machine (SVM), Random Forest (RF), and PointNet++. Despite demonstrating a F1-score weighted at 70.3%, outperforming SVM but slightly lagging RF (71.6%) and significantly trailing PointNet++ (90.3%), the proposed 1D-CNN showcases a cost-effective potential for the segmentation of *road* and *building* classes. The relative computational requirements of the models were also discussed, highlighting the practical advantages and limitations of each approach.

1. INTRODUCTION

Artificial Neural Networks (ANNs) are one of the oldest Machine and Deep Learning technologies and have been in use for over 60 years (Kurzweil, 2000). During this period, ANNs have proven to be effective in a wide range of classification and regression tasks. However, as the volume of data and task complexity increased, ANNs started encountering challenges in achieving optimal performance (Hoang et al., 2021).

Convolutional Neural Networks (CNNs) emerged as an alternative to ANNs (Chen et al., 2021). CNNs were specifically designed to operate with images and other types of spatial data. CNNs employ convolutional layers to extract relevant features from input data. This approach has proven to be highly effective in multidimensional tasks such as image classification, object detection, and semantic segmentation (Kattenborn et al., 2021).

In addition to traditional CNNs optimized for planar and spatial data, there exists a distinct variant known as 1D-CNN. Unlike their 2D and 3D counterparts, 1D-CNNs are tailored for processing one-dimensional sequential data (Kiranyaz et al., 2021), such as time series, audio signals, and text data. 1D-CNNs employ convolutional layers and other architectural adaptations specifically suited to sequential data features. This specialization offers an alternative avenue for handling value-based ML methods (including ANNs), allowing for the effective analysis and extraction of features from temporal and sequential data sources (Ozcanli and Baysal, 2022).

The objective of this work is to analyse 1D-CNN as a more effective alternative to ML classifiers when applied to point cloud processing. The research entails the selection and extraction of ten geometric point cloud features, the design, and optimization of a 1D-CNN network for semantic segmentation of an urban point cloud scanned using Mobile Laser Scanning (MLS) technology.

The reminder of this paper is structured as follows. In Section 2, works on semantic point cloud segmentation are compiled. The proposed method is explained in Section 3. Section 4 is dedicated to present the experiments. Sections 5 concludes this paper.

2. RELATED WORK

2.1 Machine Learning approaches with feature extraction

Machine learning (ML) has revolutionized, among other fields, point cloud semantic segmentation. Feature extraction in this context involves identifying relevant characteristics or attributes from these points to enable accurate object recognition and classification. Feature extraction is a process highly dependent on the prior knowledge of the developer. ML in point clouds has been employed in many applications and scenarios. In (Rashdi et al., 2023), a Support Vector Machine (SVM), a Random Forest (RF) and an Artificial Neural Network (ANN) are compared with two MLS point cloud data in an urban environment, concluding RF reaches the highest accuracies. In (Grilli et al., 2019), a RF and the OvO classifier are applied to classified the Temple of Neptune (Paestrum) in ten classes, obtaining an average of the F1-score values of each class of 91.92% with the RF and a 91.43% with the OvO classifier. Furthermore, there is more specialized research focusing on the distribution of geometric features. In Atik et al., (2021) the classification performance of different ML algorithms in multiple geometric scales was evaluated. The geometric features were generated based on the eigenvalues of the covariance matrix. Eight supervised classification algorithms were tested in four different areas from three datasets. Considering different areas, they obtained

* Corresponding author

accuracies of 93.12% with RF, 92.78% with a Multilayer Perceptron algorithm, 79.71% with SVM and a 97.30% with Linear Discriminant Analysis. In the (Balado et al., 2023), a distribution model based on a Klemperer rosette is contrasted with conventional approaches that rely on search radius and the number of nearest neighbours, concluding that the information provided by the nearest points is the most influential and significant for the given context.

2.2 Deep Learning approaches with feature extraction

Although Deep Learning (DL) applications tend to be thought of as an end-to-end approach, there are numerous applications that implement a feature extraction phase to guide the NN during training and prediction. Exhaustive review on DL with 3D data can be found in (Liu et al., 2019; Bello et al., 2020; Guo et al., 2021). DL with feature extraction approach offers an efficient compromise, initiating from 3D data while enabling the extraction of complex features and the utilization of NN architectures originally designed for different data formats. In (Paz Mouriño et al., 2021), geometric features are harnessed to produce synthetic colour images derived from MLS urban data. These images are then subjected to semantic segmentation using a 2D CNN Reset. In (Balado et al., 2020), rasterized images of the point cloud are integrated with images of real-world objects obtained from Google Images to facilitate the classification of indoor furniture. Nonetheless, as in ML, the efficacy of feature extraction remains contingent upon the developer's expertise and domain knowledge.

2.3 End-to-end DL approaches

After the first architecture designed specifically for point cloud, PointNet (Qi et al., 2017a), other approaches have proven effective in processing and extracting valuable insights from these complex point cloud datasets, such as RandLA-Net (Hu et al., 2020) or PointCNN (Li et al., 2018). In the context, a DL framework for point cloud segmentation was introduced by (Pierdicca et al., 2020). This framework enhances the Dynamic Graph Convolutional Neural Network (DGCNN) by incorporating meaningful features such as normals and colours. The authors applied this method to Architectural Cultural Heritage and compared it with other neural networks, including DGCNN (Wang et al., 2019), PointNet (Qi et al., 2017a), PoinNet++ (Qi et al., 2017b) and PCNN (Johnson and Padgett, 1999). For the semantic segmentation of industrial point clouds, (Yin et al., 2021) proposed a new deep learning-based approach called ResPointNet++. Their methodology involved a dataset comprising 80 million points, labelled with five classes. Remarkably, they achieved an overall segmentation accuracy of 94% and a mean Intersection over Union (mIoU) of 87%, surpassing the performance of PointNet++. Nevertheless, end-to-end DL approaches necessitate extensive training datasets, which may not be easily obtainable in some environments, e.g. train stations (Lumban-Gaol et al., 2021).

2.4 Contribution

In this work, a DL not end-to-end solution is presented and studied with an initial stage of feature extraction followed by the design and implementation of a 1D CNN to relate complex features. Furthermore, the proposed 1D CNN will be compared with other state-of-the-art ML modes (SVM and RF) and with end-to-end approach based on PointNet++.

3. MATERIALS AND METHODS

3.1 Data

The data utilized in this study originate from the Santiago Urban Dataset (González-Collazo et al., 2024). The urban area was scanned using an MLS Riegl VUX-1HA. Street K was used for training and validation purposes. Due to the imbalanced nature of the class distribution, after feature computation, 20000 samples per class were randomly selected for training and validation. Street F and M were entirely reserved for testing. The target classes encompassed 8 categories: *road*, *sidewalk*, *curb*, *building*, *vehicle*, *vegetation*, *pole*, and *other*.

3.2 Feature extraction

The number of features defines the size of the input layer of the 1D CNN. Recent works have shown that using too many features can be counterproductive for the AI algorithm (Weinmann and Weinmann, 2019), so in this work only 10 features related to the eigenvalues, the surface normals and radius are selected with a neighbourhood k = 25 nearest neighbours.

To calculate eigenvalues from a point cloud $P(P_X P_Y P_Z)$ the covariance matrix S is needed. Covariance matrix summarizes how the data points vary together in three-dimensional space. The covariance between two points P_i with $P_X P_Y P_Z$ coordinates is calculated as:

$$S = \frac{1}{k+1}\sum_{i=0}^{K}(P_i - \bar{P})(P_i - \bar{P})^T = 0 \quad (1)$$

With the covariance matrix S, the eigenvalues (λ) are calculated following equation (2). The three eigenvalues represent the spread of data along the three principal axes of variation in the point cloud. The eigenvalues are typically sorted in descending order, so $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$.

$$det(S - \lambda \mathrm{I}) = 0 \quad (2)$$

From the eigenvalues $(\lambda_1, \lambda_2, \lambda_3)$, the surface normals (N_X, N_Y, N_Z) and the radius r, the following 10 features are estimated (Weinmann et al., 2015) with equations (3 to 12):

$$anisotropy\ A_\lambda = \frac{\lambda_1 - \lambda_3}{\lambda_1} \quad (3)$$

$$eigentropy\ E_\lambda = -\sum_{i=1}^{3} e_i \ln(e_i) \quad (4)$$

$$eigensum\ \Sigma_\lambda = \lambda_1 + \lambda_2 + \lambda_3 \quad (5)$$

$$curvature\ C_\lambda = \frac{\lambda_3}{(\lambda_1 + \lambda_2 + \lambda_3)} \quad (6)$$

$$omnivariance\ O_\lambda = \sqrt[3]{\lambda_1 \lambda_2 \lambda_3} \quad (7)$$

$$linearity\ L_\lambda = \frac{\lambda_1 - \lambda_2}{\lambda_1} \quad (8)$$

$$planarity\ P_\lambda = \frac{\lambda_2 - \lambda_3}{\lambda_1} \quad (9)$$

$$sphericity\ S_\lambda = \frac{\lambda_3}{\lambda_1} \quad (10)$$

$$verticality\ V_N = 1 - |N_Z| \quad (11)$$

$$density\ D_r = \frac{3(k+1)}{4\pi r^3} \quad (12)$$

3.3 1D CNN architecture definition and training

In the examination of 1D CNN architectures, certain hyperparameters are held constant across all tests. This decision is grounded in the extensive research conducted by the authors in previous works, which has established nearly standardized values for many hyperparameters. The unchanging hyperparameters are itemized in Table 1 for reference.

Hyperparameter	Value
Learning rate	0.001
Optimizer	Adam
Regularization method	Dropout
Dropout	0.3
Activation method (conv)	RelU
Kernel size (conv)	3
Kernel size (pool)	2
Strides (conv)	1
Strides (pool)	2

Table 1. Fixed hyperparameters.

The optimization effort of the CNN architecture focuses on the selection of the number of convolutional layers, neurons, pooling layers and dropouts, and on the training (number of epochs and batch size). The selection was experimentally conducted by training and validating the network with 5000 samples per class on the weighted F1-score results.

3.3.1 The convolutional layer plays a pivotal role in reducing the number of parameters within the neural network while efficiently extracting essential features. In this layer, a small weight matrix, referred to as a kernel or filter, is systematically moved across the input data. At each position during this process, a computation involving multiplication and summation takes place within specific regions of the feature map. The outcome of these operations yields a novel feature map that accentuates distinct patterns or attributes within the 1D input vector. The convolutional layers also govern the network's depth and its ability to capture intricate data relationships. The number of selected convolutional layers is 8, as it yielded the highest F1-weighted score (Figure 1).

3.3.2 The number of neurons in each layer (hidden units) governs the complexity and the network's capacity to capture patterns in the data. The convolutional layers have 256 and 512 neurons, as determined by the configuration that resulted in the highest F1-weighted score (Figure 2).

3.3.3 The pooling layer is employed to diminish the spatial dimensions of the input representation, consequently reducing the quantity of parameters and computational workload in the network. This reduction layer applies an aggregation function over a rectangular region of the input, using the maximum function (max-pooling) in this case, to derive a single value representing that specific area. Three pooling layers have been selected, as they yielded a slightly higher F1-weighted score compared to other configurations (Figure 3).

3.3.4 The dropout layer involves randomly deactivating a fraction of neurons during each training iteration, to identify redundant patterns in the data, thereby mitigating overfitting. Dropout has proven to be effective across a spectrum of tasks and neural network architectures, including CNNs, owing to its enhanced resilience and adaptability. Its ability to enhance generalization and overfitting has been validated in numerous applications (Srivastava et al., 2014). In this architecture, only one dropout layer is chosen, as it delivers the highest F1-weighted score (Figure 4).

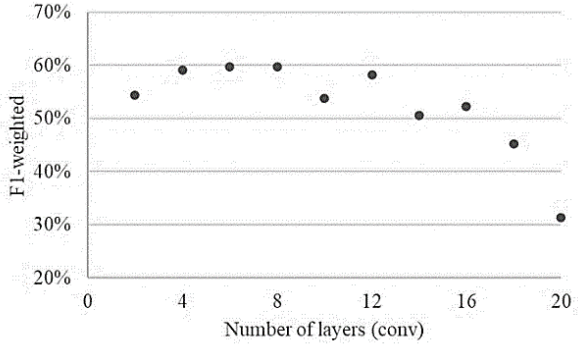

Figure 1. F1-score according to number of convolutional layers.

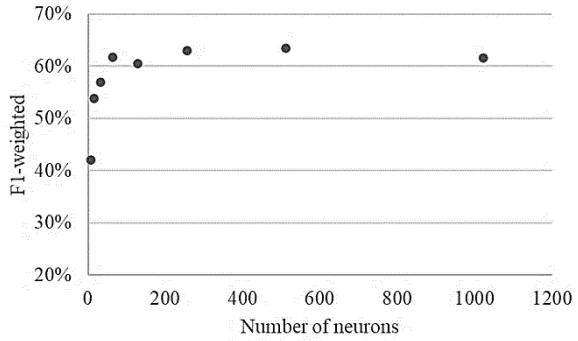

Figure 2. F1-score according to number of neurons.

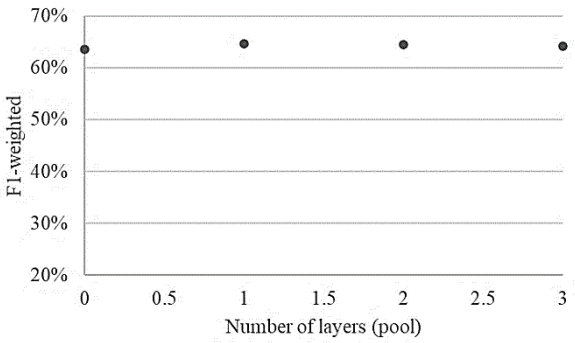

Figure 3. F1-score according to number of pooling layers.

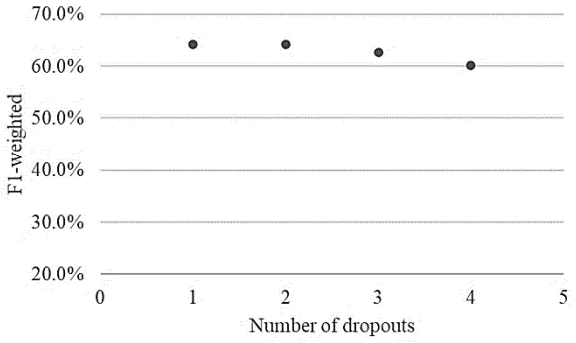

Figure 4. F1-score according to number of dropouts.

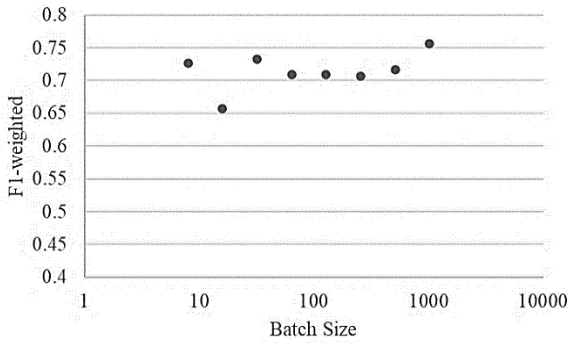

Figure 5. F1-score according to batch size for 70 epochs.

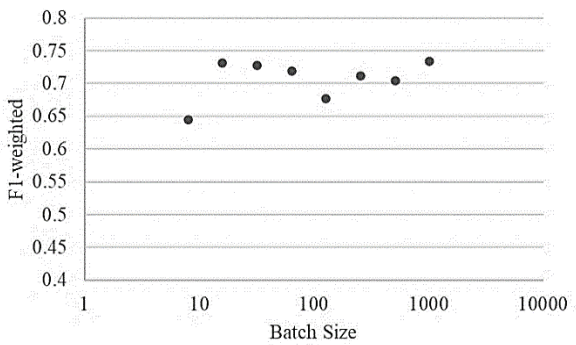

Figure 6. F1-score according to batch size for 100 epochs.

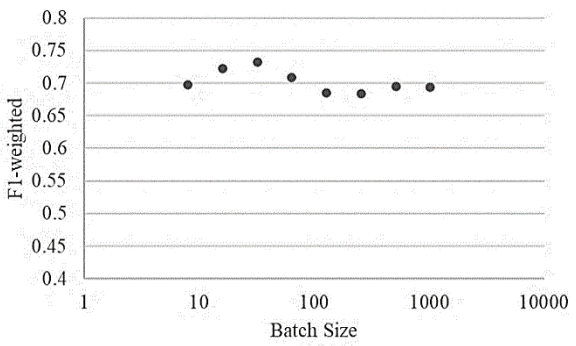

Figure 7. F1-score according to batch size for 200 epochs.

3.3.5 **The dense layer**, also known as a fully connected layer, is a type of layer in which each node or unit in the previous layer is connected to every node in the current layer, receiving input from all the outputs of the previous layer. The dense layer is particularly valuable for classification and prediction tasks because it enables the neural network to discover nonlinear relationships between input data and output labels. The dense layer is positioned at the end of the CNN architecture and comprises as many neurons as there are predicted classes.

3.3.6 To determine the **number of epochs and batch size**, a series of training experiments with their respective validations are conducted, varying the number of epochs (up to 200) and batch size (ranging from 8 to 1024). Training runs with a low number of epochs were disregarded due to their inferior performance when compared to the results obtained for 70 epochs (Figure 5), 100 epochs (Figure 6), and 200 epochs (Figure 7). An excessively high number of epochs leads to a notable increase in processing time without a significant improvement in the F1-weighted score. Furthermore, the batch size refers to the quantity of samples that can be concurrently processed by the CNN and is constrained by the available computer memory (RAM or GPU). Given these considerations, a compromise solution was reached by selecting 100 epochs for the final training and a batch size of 64, even though it may not yield the highest F1-weighted score configuration.

Table 2 enumerates the architecture and hyperparameters derived from empirical experimentation.

Hyperparameter	Value
Layers (conv)	4+4
Neurons	256-512
Layers (pool)	3
Layer (dropout)	1
Epochs	100
Batch	64

Table 2. Experimental hyperparameters.

4. RESULTS AND ANALYSIS

4.1 Training performance

Table 3 presents the training results on the validation set. Both the SVM and the 1D CNN exhibited underfitting and were unable to capture the complexities of the urban environment data. This underfitting is primarily attributed to the simplicity of the SVM and the 1D CNN, as the RF managed to achieve a better fit with the same samples. Additionally, an increase in the number of samples leaded to an improvement in the results of the SVM, but not of the 1D CNN, as no significant increase was observed when transitioning from 5000 to 20000 samples per class. Conversely, the RF displayed clear overfitting (reaching 100% F1-score), which will be confirmed in the following Section 4.2, because RF exhibited a F1-score reduction on the test set.

	F1-weighted
SVM	40.0%
1D CNN	68.8%
RF	100%

Table 3. Validation results during training.

	Road	*Sidewalk*	*Curb*	*Building*	*Vehicle*	*Vegetation*	*Pole*	*Other*	Weighted
SVM	58.1%	19.9%	1.4%	30.1%	0.0%	37.9%	2.3%	0.0%	**31.8%**
1DCNN	76.6%	53.6%	4.8%	77.2%	45.3%	47.1%	8.6%	5.5%	**70.3%**
RF	82.9%	51.7%	4.7%	77.2%	46.7%	47.7%	9.4%	5.4%	**71.6%**
PointNet++	90.8%	75.6%	64.1%	95.0%	83.1%	81.2%	57.1%	23.3%	**90.3%**

Table 4. F1-score comparison in the test set by class and weighted by the number of samples.

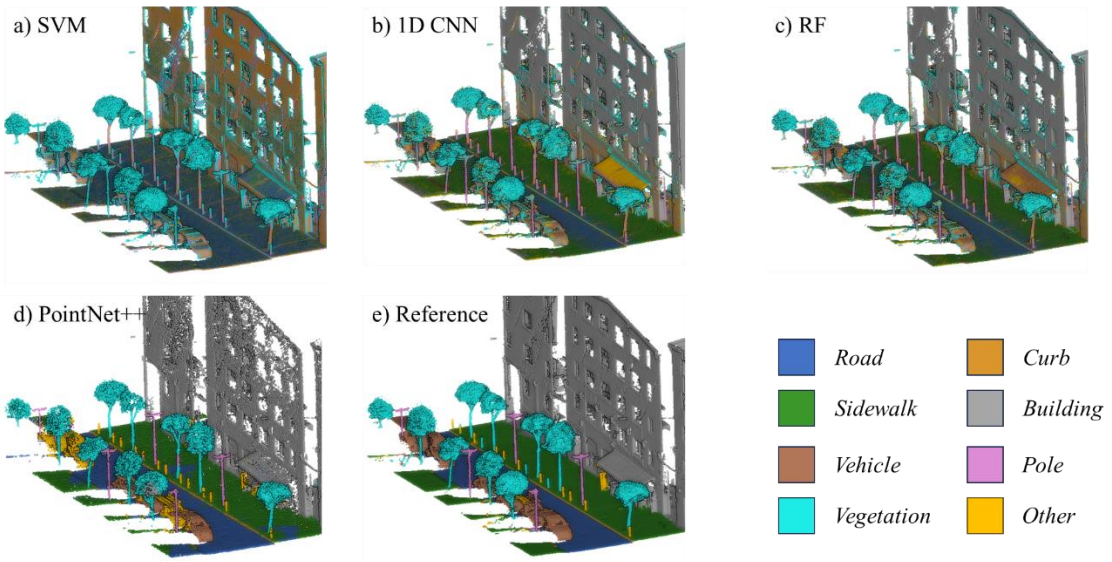

Figure 8. Street F on test set coloured by class (one facade removed to improve visualization).

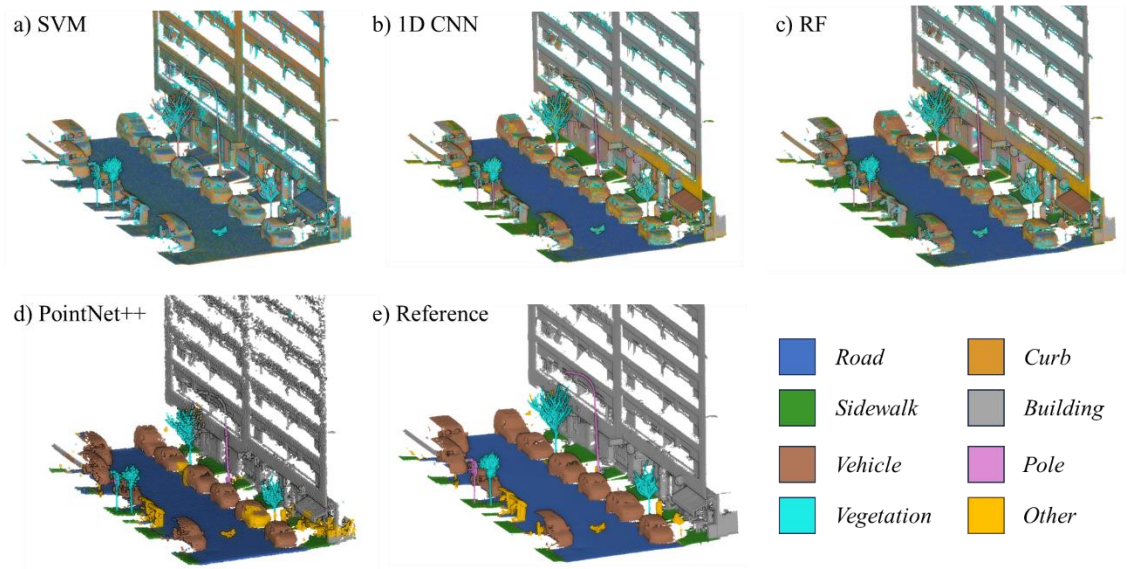

Figure 9. Street M on test set coloured by class (one facade removed to improve visualization).

4.2 Quantitative and qualitative comparison

The comparison of F1-scores per class for the application of SVM, RF, 1D CNN, and PointNet++ is illustrated in Table 4, alongside the F1-score weighted with the sample number. Similar F1-score values were observed between 1D CNN and RF, despite the substantial disparities in the training. SVM achieved a notably low F1-weighted (31.8%). PointNet++ attained the best result with a F1-weighted of 90.3%. However, PointNet++ was trained with all the SUD samples, except those of test, without limiting itself to 20000 samples per class as in the other classifiers.

The performance of 1D CNN and RF was highly comparable, even identical for the *buildings* class. However, 1D CNN exhibited superior performance exclusively in segmenting *sidewalks* and *curbs*, whereas RF excelled in the segmentation of *roads*, *vehicles*, *vegetation*, and *poles*. Overall, the behaviour of 1D CNN displayed typical traits of ML algorithms, with outcomes heavily favouring classes with larger sample sizes over those with fewer samples. Notably, PointNet++ demonstrated significantly superior results across all classes.

Figures 8 and 9 depict segments of streets F and M, utilized during the testing phase. Substantial disparities in the point distribution between the 1D CNN and RF results were not discernible. Both models exhibit confusions primarily pertaining to the accurate differentiation between *road* and *sidewalk* segments, misclassification of tree trunks as *poles*, the erroneous classification of lower building sections as *other* entities, and the misattribution of *vegetation* points to the building boundaries.

4.3 Computing time

Two different computers were used for training and testing:

- Computer 1: CPU Intel Core i5-1135G7 with 8GB RAM and GPU Intel Iris Xe Graphics.
- Computer 2: CPU Intel Core i7-10750H with 16GB RAM and GPU Nvidia RTX2060 Mobile.

The training durations are documented in Table 5. Notably, the SVM model exhibited the lengthiest training period, coinciding with its comparatively suboptimal performance outcomes. Conversely, the RF and 1D CNN models featured similar training times. PointNet++ model necessitated a longer training period, spanning a total of 24 hours. However, this extended training duration was conducted on a significantly more potent computational infrastructure and with an expanded dataset sample size. The RF and 1D CNN exhibited efficient training times and relatively comparable performance while PointNet++, while requiring a more extended training period, showcased its capabilities when executed on a substantially powerful computing platform and a larger dataset.

Computer	Samples	AI Model	Training time
1	160k	SVM	40 hours
1	160k	RF	10 hours
1	160k	1D CNN	9.4 hours
2	143M	PointNet++	24 hours

Table 5. Training times.

4.4 Discussion

Although promising, the results obtained with the 1D CNN did not meet expectations and even failed to outperform RF, a benchmark Machine Learning technique. While the proposed 1D CNN demonstrated effective segmentation capabilities for *building* and *road* classes, slightly better values were achieved with RF, without the need to invest time in configuring the numerous parameters required for 1D CNN optimization. Additionally, training times of 1D CNN and RF were similar.

Regarding the comparison between the 1D CNN and PointNet++, the 1D CNN (and the RF) demonstrated the feasibility of training on computers with limited computational resources. In contrast, the implementation of PointNet++ required more robust computational (graphic) resources and a longer training time. However, the results obtained with PointNet++ were close to the reference data, affirming that point Neural Networks are more effective for accurately segmenting point clouds in urban areas than other ML methods.

5. CONCLUSION AND FUTURE WORK

In this work, a 1D Convolutional Neural Network was designed for segmenting urban point clouds acquired through Mobile Laser Scanning. Ten well-known local geometric features of point clouds were used as input for the 1D CNN. Optimal numbers of convolution layers, neurons, pooling layers, dropout layers, dense layers, training epochs, and batch size were empirically selected. The proposed 1D CNN was trained and tested using data from the Santiago Urban Dataset, and compared with Support Vector Machine, Random Forest, and PointNet++.

The 1D CNN exhibited a F1-score weighted of 70.3%, notably superior to SVM (31.8%), slightly lower than RF (71.6%), and significantly behind PointNet++ (90.3%), even though PointNet++ was trained with a much larger dataset. The 1D CNN proved to be a cost-effective solution for segmentation, primarily for *road* and *building* classes, compared to end-to-end Deep Learning methods such as PointNet++, but RF also demonstrated similar behaviour without require tedious parameter selection.

Future work will consider the use of post-processing techniques, such as morphological filtering, to enhance the coherence and precision of segmentations, particularly in areas where both the 1D CNN and RF exhibit weaknesses.

ACKNOWLEDGEMENTS

This work was partially supported by Xunta de Galicia ED481D-2023-005, and Government of Spain through project PID2019-105221RB-C43 funded by MCIN/AEI/10.13039/501100011033 and RYC2022-038100-I funded by MCIN/AEI/10.13039/501100011033 and FSE+.

REFERENCES

Atik, M.E., Duran, Z., Seker, D.Z., 2021. Machine Learning-Based Supervised Classification of Point Clouds Using Multiscale Geometric Features. ISPRS International Journal of Geo-Information 10. https://doi.org/10.3390/ijgi10030187

Balado, J., Díaz-Vilariño, L., Verbree, E., Arias, P., 2020. Transfer Learning for indoor object classification: from images to point clouds. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences V-4–2020, 65–70. https://doi.org/10.5194/isprs-annals-V-4-2020-65-2020

Balado, J., Fernández, A., González, E., Díaz-Vilariño, L., 2023. Semantic Point Cloud Segmentation Based on Hexagonal Klemperer Rosette and Machine Learning, in: Cavas-Martínez, F., Marín Granados, M.D., Mirálbes Buil, R., de-Cózar-Macías, O.D. (Eds.), Advances in Design Engineering III. Springer International Publishing, Cham, pp. 617–629.

Bello, S.A., Yu, S., Wang, C., Adam, J.M., Li, J., 2020. Review: Deep Learning on 3D Point Clouds. Remote Sensing 12. https://doi.org/10.3390/rs12111729

Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., Miao, Y., 2021. Review of Image Classification Algorithms Based on Convolutional Neural Networks. Remote Sensing 13. https://doi.org/10.3390/rs13224712

González-Collazo, S.M., Balado, J., Garrido, I., Grandío, J., Rashdi, R., Tsiranidou, E., Río-Barral, P. del, Rúa, E., Puente, I., Lorenzo, H., 2024. Santiago urban dataset SUD: Combination of Handheld and Mobile Laser Scanning point clouds. Expert Systems with Applications 238, 121842. https://doi.org/10.1016/j.eswa.2023.121842

Grilli, E., Özdemir, E., Remondino, F., 2019. Application of Machine and Deep Learning Strategies for the Classification of Heritage Point Clouds. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-4/W18, 447–454. https://doi.org/10.5194/isprs-archives-XLII-4-W18-447-2019

Guo, Y., Wang, H., Hu, Q., Liu, H., Liu, L., Bennamoun, M., 2021. Deep Learning for 3D Point Clouds: A Survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 43, 4338–4364. https://doi.org/10.1109/TPAMI.2020.3005434

Hoang, A.T., Nižetić, S., Ong, H.C., Tarelko, W., Pham, V.V., Le, T.H., Chau, M.Q., Nguyen, X.P., 2021. A review on application of artificial neural network (ANN) for performance and emission characteristics of diesel engine fueled with biodiesel-based fuels. Sustainable Energy Technologies and Assessments 47, 101416. https://doi.org/10.1016/j.seta.2021.101416

Hu, Q., Yang, B., Xie, L., Rosa, S., Guo, Y., Wang, Z., Trigoni, N., Markham, A., 2020. RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds.

Johnson, J.L., Padgett, M.L., 1999. PCNN models and applications. IEEE Transactions on Neural Networks 10, 480–498. https://doi.org/10.1109/72.761706

Kattenborn, T., Leitloff, J., Schiefer, F., Hinz, S., 2021. Review on Convolutional Neural Networks (CNN) in vegetation remote sensing. ISPRS Journal of Photogrammetry and Remote Sensing 173, 24–49. https://doi.org/10.1016/j.isprsjprs.2020.12.010

Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J., 2021. 1D convolutional neural networks and applications: A survey. Mechanical Systems and Signal Processing 151, 107398. https://doi.org/10.1016/j.ymssp.2020.107398

Kurzweil, R., 2000. The age of spiritual machines: When computers exceed human intelligence. Penguin.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. PointCNN: Convolution On X-Transformed Points, in: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (Eds.), Advances in Neural Information Processing Systems. Curran Associates, Inc.

Liu, W., Sun, J., Li, W., Hu, T., Wang, P., 2019. Deep Learning on Point Clouds and Its Application: A Survey. Sensors 19. https://doi.org/10.3390/s19194188

Lumban-Gaol, Y.A., Chen, Z., Smit, M., Li, X., Erbaşu, M.A., Verbree, E., Balado, J., Meijers, M., van der Vaart, N., 2021. A comparative study of point clouds semantic segmentation using three different neural networks on the railway station dataset. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLIII-B3-2021, 223–228. https://doi.org/10.5194/isprs-archives-XLIII-B3-2021-223-2021

Ozcanli, A.K., Baysal, M., 2022. Islanding detection in microgrid using deep learning based on 1D CNN and CNN-LSTM networks. Sustainable Energy, Grids and Networks 32, 100839. https://doi.org/10.1016/j.segan.2022.100839

Paz Mouriño, S. de, Balado, J., Arias, P., 2021. Multiview Rasterization of Street Cross-sections Acquired with Mobile Laser Scanning for Semantic Segmentation with Convolutional Neural Networks, in: IEEE EUROCON 2021 - 19th International Conference on Smart Technologies. pp. 35–39. https://doi.org/10.1109/EUROCON52738.2021.9535645

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. https://doi.org/10.3390/rs12061005

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017a. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017b. PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space.

Rashdi, R., Balado, J., Sánchez, J.M., Arias, P., 2023. Comparative Study of Road and Urban Object Classification Based on Mobile Laser Scanners. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLVIII-1/W1-2023, 423–429. https://doi.org/10.5194/isprs-archives-XLVIII-1-W1-2023-423-2023

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15, 1929–1958.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic Graph CNN for Learning on Point Clouds. ACM Trans. Graph. 38. https://doi.org/10.1145/3326362

Weinmann, M., Jutzi, B., Hinz, S., Mallet, C., 2015. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. ISPRS Journal of Photogrammetry and Remote Sensing 105, 286–304. https://doi.org/10.1016/j.isprsjprs.2015.01.016

Weinmann, M., Weinmann, M., 2019. Fusion of hyperspectral, multispectral, color and 3D point cloud information for the semantic interpretation of urban environments. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XLII-2/W13, 1899–1906. https://doi.org/10.5194/isprs-archives-XLII-2-W13-1899-2019

Yin, C., Wang, B., Gan, V.J.L., Wang, M., Cheng, J.C.P., 2021. Automated semantic segmentation of industrial point clouds using ResPointNet++. Automation in Construction 130, 103874. https://doi.org/10.1016/j.autcon.2021.103874