

---

# Semantic Classification of 3D point clouds with multiscale spherical neighborhoods

NPM3D project report

---

Inès Vati\*

MVA, ENS Paris-Saclay, Cachan, France

[ines.vati@eleves.enpc.fr](mailto:ines.vati@eleves.enpc.fr)

## Abstract

In this work, I studied the methodology presented by Thomas et al. [5] for classifying each point within 3D point clouds using an innovative feature design approach. A series of experiments are conducted to assess the performance of the method and to compare it with other approaches ranging from neighborhood definition to classification algorithm, through the use of additional features. As no code were provided, the methods were implemented from scratch. The code implemented for this project is available on <https://github.com/InesVATI/npm3d-project>.

**Keywords.** 3D point clouds, semantic classification, multiscale, spherical neighborhoods, Random Forest Classifier

## 1 Proposed method

In their study [5], the authors employ a multiscale approach to design informative features, drawing upon the findings of Hackel et al. in 2016 [2], which demonstrated the improved efficiency of such an approach over adapting the scale for each point [7]. To compute these features, the authors adopt a spherical definition of neighborhoods, incorporating neighbors within a radius that exponentially increases with the scale  $s$ , denoted as  $r_s = r_0 * \phi^s$ . This formulation ensures that the neighborhoods appropriately reflect the density of the point cloud.

Let's consider a set of  $N$  points of interest for which we aim to compute the multiscale features. For each scale, the same set of  $N_{feats}$  features is computed, as detailed in Table 1.  $S$  is the number of scales. The feature computation proceeds through the following steps:

1. At scale 0, the eigenvalues and eigenvectors of the covariance matrix of the neighborhood, obtained with an initial radius  $r_0$ , are computed. Subsequently, the features for each point of interest are derived using the formulas outlined in Table 1.
2. For each scale  $s$ , a grid subsampling is applied to the original cloud using a voxel size of  $r/\rho$ . Each original point is then assigned to the closest voxel center, inheriting the features of the voxel center for that particular scale.
3. The  $N_{feats}$  features are computed for each voxel center associated with at least one point of interest from the original point cloud, within a spherical neighborhood of radius  $r_s = r_0 * \phi^s$ .
4. Consequently, we obtain a feature matrix of size  $N \times N_{feats} * S$ .

Here,  $\phi$  is the ratio between the radius of consecutive neighborhoods and  $\rho$  is the ratio between the radius of the spherical neighborhood and the voxel size of the grid subsampling.  $\rho$  enables to control the maximum number of subsampled points that a neighborhood can contain.

---

\*Engineering student from École des Ponts ParisTech, Champs-sur-Marne, France

Feature	Formula
Sum of eigenvalues	$\sum \lambda_i$
Omnivariance	$(\prod \lambda_i)^{(1/3)}$
Eigenentropy	$-\sum \lambda_i \ln(\lambda_i)$
Linearity	$\frac{\lambda_1 - \lambda_2}{\lambda_1}$
Planarity	$\frac{\lambda_2 - \lambda_3}{\lambda_1}$
Sphericity	$\frac{\lambda_3}{\lambda_1}$
Change of curvature	$\lambda_1 - \lambda_3$
Verticality ( $\times 2$ )	$ \arcsin(\langle e_i, e_z \rangle) _{i=1,3}$
Absolute moment ( $\times 6$ )	$\frac{1}{ \mathcal{N} }  \sum_{j \in \mathcal{N}} \langle p_j - \mathbf{p}, e_i \rangle^k _{k=1,2; i=1,2,3}$
Vertical moment ( $\times 2$ )	$\frac{1}{ \mathcal{N} }  \sum_{j \in \mathcal{N}} \langle p_j - \mathbf{p}, e_z \rangle^k _{k=1,2}$
Number of points	$ \mathcal{N} $

**Table 1 :**  $N_{feats} = 18$  Geometric features [5] of point  $\mathbf{p}$ .  $\lambda_i$  are the eigenvalues of the covariance matrix of the neighborhood  $\mathcal{N}$  of  $\mathbf{p}$ , sorted in decreasing order.  $e_i$  are the corresponding eigenvectors.

Feature	Formula
Vertical range	$z_{max} - z_{min}$
Height below	$\mathbf{p}_z - z_{min}$
Height above	$z_{max} - \mathbf{p}_z$

**Table 2 :** Additional height features from [2, 4], where  $z_{max} = \max_{j \in \mathcal{N}} p_{j,z}$  and  $z_{min} = \min_{j \in \mathcal{N}} p_{j,z}$ .  $\mathcal{N}$  is the neighborhood of point  $\mathbf{p}$ .  $p_{j,z}$  is the coordinate of point  $j$  in the vertical axis.

## 2 Experiments and results

I used two datasets to evaluate the method: the Paris-rue-Cassette dataset<sup>2</sup>, a point cloud of 12 million points and the NPM3D dataset<sup>3</sup>, that groups three point clouds, including the MiniLille and MiniParis point clouds. The MiniLille dataset has 1,901,853 and 2,500,428 points and the MiniParis dataset has 4,159,318 points.

The Paris-rue-Cassette dataset serves as the basis for comparing various methods. After carefully selecting parameters for a Random Forest classifier, I applied the same classifier configuration to investigate the following:

- The predictive performance, comparing the use of a spherical neighborhood versus a K-Nearest Neighbors (KNN) neighborhood (detailed in Section 2.2).
- The impact of using multiple scales, comparing the proposed method with and without multiscaling (outlined in Section 2.3).
- The significance of features and the potential benefits of incorporating additional height features, as suggested by Mohamed et al. [4] (explored in Section 2.4).

For all the experiments, the tree classifier employed 150 trees and the Gini criterion. In addition, balanced class weights were used to address class imbalances, with weights being inversely proportional to class frequencies in the input data. Notably, no maximum depth constraint were set, as this led to better results across all approaches. Consequently, nodes were expanded until either all leaves were pure or until each leaf contained fewer than two samples.

<sup>2</sup><http://data.ign.fr/benchmarks/UrbanAnalysis/>

<sup>3</sup><https://npm3d.fr/benchmark-for-master-course-on-3d-point-clouds>

Furthermore, the NPM3D dataset is used in Section 2.5 to assess the method’s generalization ability to datasets not involved in the classifier training.

To assess the performance of the different approaches, the Intersection over Union (IoU) metric or jaccard index is used. It is given by

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}$$

where TP is the number of true positive, FP the number of false positive and FN the number of false negative. I also computed the weighted IoU which is the mean of the IoU for each class, weighted by the number of points of each class. As performed in [5], the experiments were repeated 10 times and the mean and standard deviation of the Intersection over Union (IoU) metric were computed and are presented in Tables 4 and 5.

## 2.1 Paris-rue-cassette dataset

The raw ground truth contained a large number of classes. I had to parse an XML file to group together some label classes to fairly compare the results with those obtained in previous studies [5, 2, 7].

Due to computational constraints, I applied grid subsampling to reduce the size of the point cloud, using a voxel size of 0.1m. The resulting point cloud contains 1,288,215 points. Each voxel center was assigned a label based on the majority label of the points within the voxel. Notably, some classes suffered a significant reduction in the number of points, such as the ‘Traffic Sign’ label. The number of points per class is given in Table 3. While it would have been possible to retain all points from these underrepresented classes in the subsampled point cloud, it is pertinent to explore strategies for managing class imbalances, as discussed in Section 3.

Label	Size
Ground	188,644
Building	954,267
Traffic Signs	195
Pedestrians	2,079
Cars	28,437
Vegetation	98,175
Motorcycles	4,523

**Table 3 :** Class size in Paris-rue-Cassette subsampled cloud

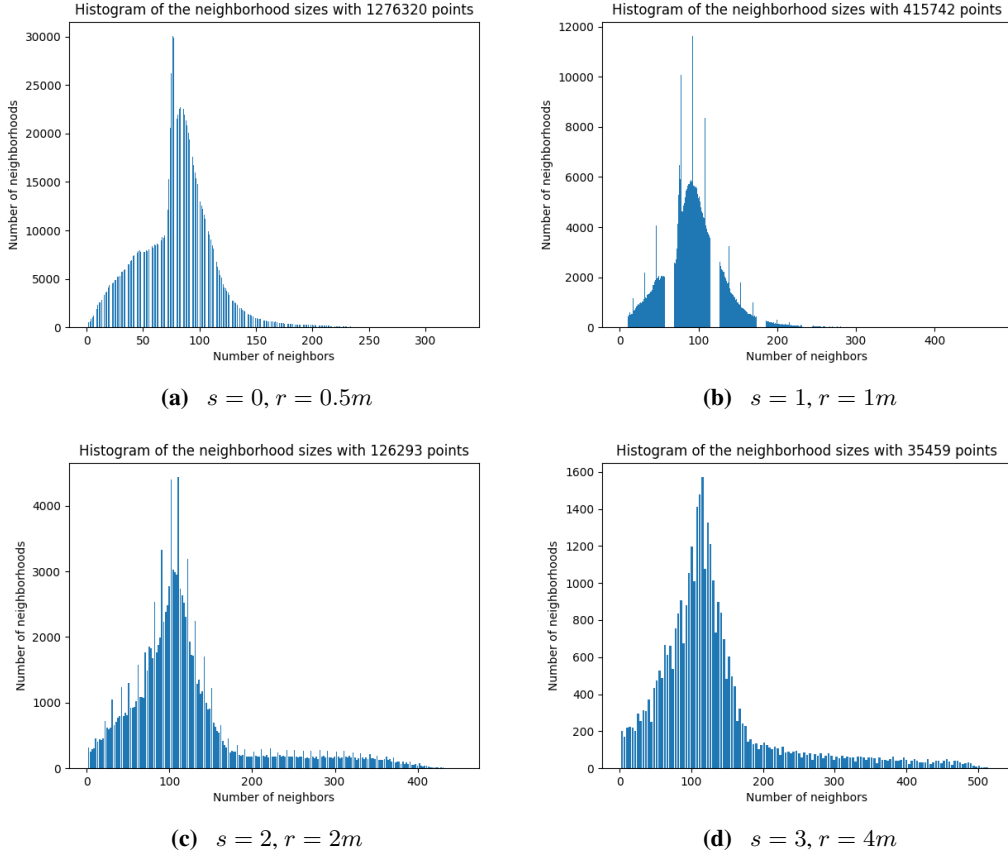
On this dataset, I choose  $S = 4$ ,  $r_0 = 0.5\text{m}$ ,  $\phi = 2$ ,  $\rho = 5$ . The figure 1 displays the distribution of the neighborhood sizes in the subsampled cloud at the different scale to check that the neighborhoods are not too small with those parameters. One can notice that some points have very small neighborhoods even without subsampling. However, the neighborhood size is 100 points in average. We can also see that the iterative grid subsampling do not distort too much the neighborhood size distribution. Choosing  $S = 4$  number of scales seems to be an appropriate choice to avoid having empty neighborhoods while computing the covariance matrix.

It is worth noting that the radius of the spherical neighborhood adjusts according to the scale, ensuring that the neighborhood consistently represents a fixed portion of space at every scale. This differs from KNN neighborhood definition, where the neighborhood size is fixed, which can result in an inaccurate description of the neighborhood, particularly at smaller scales.

## 2.2 Comparing Neighborhood Definitions

By employing spherical neighborhoods instead of KNN, the features represent a consistent portion of space at each scale.

The scores presented in Table 4 shows that the use of spherical neighborhoods consistently outperforms the KNN neighborhood definition. For instance, the ‘Traffic Signs’ class, the spherical neighborhood approach improved the Jaccard index by nearly 87% over the KNN neighborhood



**Figure 1 :** Histogram of the number of points in the neighborhood of each point for different scales  $s$  on the subsampled Paris-rue-Cassette Dataset.

Method	Ground	Building	Traffic Signs	Pedestrians	Cars	Vegetation	Motorcycles	Weighted IoU
[5]	81.48 $\pm$ 2.95	71.21 $\pm$ 8.07	90.25 $\pm$ 4.21	92.81 $\pm$ 2.00	47.48 $\pm$ 12.85	<b>82.34</b> $\pm$ 4.68	<b>97.31</b> $\pm$ 2.27	78.87 $\pm$ 4.22
+ height	<b>83.16</b> $\pm$ 2.23	<b>72.10</b> $\pm$ 3.60	<b>97.00</b> $\pm$ 2.58	<b>93.56</b> $\pm$ 1.42	<b>49.84</b> $\pm$ 12.70	81.46 $\pm$ 6.48	95.23 $\pm$ 2.99	<b>79.37</b> $\pm$ 2.95
- multiscale	68.76 $\pm$ 5.77	43.25 $\pm$ 3.03	9.08 $\pm$ 3.18	71.19 $\pm$ 1.91	30.12 $\pm$ 4.14	62.53 $\pm$ 2.57	54.84 $\pm$ 2.71	54.74 $\pm$ 1.51
knn	71.87 $\pm$ 8.28	57.21 $\pm$ 6.39	48.24 $\pm$ 27.04	90.99 $\pm$ 1.98	36.52 $\pm$ 8.29	80.76 $\pm$ 1.66	86.94 $\pm$ 1.52	70.53 $\pm$ 3.13

**Table 4 :** Average IoU (with standard deviation) on the subsampled Paris-rue-Cassette dataset using Random Forest classifier.

Method	Ground	Building	Traffic Signs	Pedestrians	Cars	Vegetation	Motorcycles	Weighted IoU
[5]	83.34 $\pm$ 2.82	69.72 $\pm$ 5.91	58.96 $\pm$ 24.93	93.37 $\pm$ 2.36	<b>60.56</b> $\pm$ 9.58	77.72 $\pm$ 6.03	96.04 $\pm$ 2.97	79.95 $\pm$ 2.73
+ height	<b>84.94</b> $\pm$ 2.38	<b>74.30</b> $\pm$ 6.57	<b>82.29</b> $\pm$ 17.82	<b>94.10</b> $\pm$ 1.51	58.91 $\pm$ 12.37	79.13 $\pm$ 5.79	<b>97.68</b> $\pm$ 1.29	<b>81.52</b> $\pm$ 3.56
- multiscale	62.13 $\pm$ 8.48	41.47 $\pm$ 3.51	9.06 $\pm$ 1.30	71.43 $\pm$ 3.35	28.87 $\pm$ 4.41	66.50 $\pm$ 1.71	59.74 $\pm$ 3.04	54.64 $\pm$ 2.44
knn	71.06 $\pm$ 9.34	56.08 $\pm$ 8.58	37.02 $\pm$ 26.41	93.68 $\pm$ 2.03	49.68 $\pm$ 13.76	<b>81.21</b> $\pm$ 3.65	91.35 $\pm$ 3.19	73.54 $\pm$ 2.73

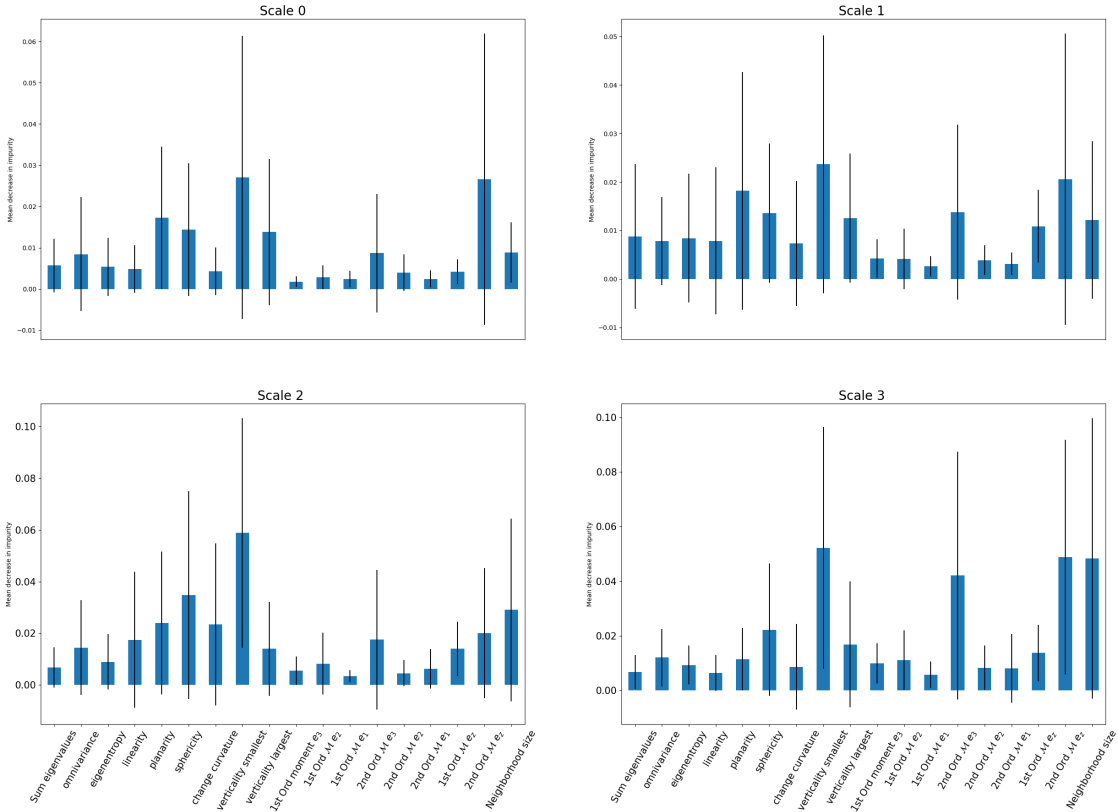
**Table 5 :** Average IoU (with standard deviation) on the subsampled Paris-rue-Cassette dataset using HBG classifier

method. Moreover, the overall weighted IoU is 18% higher with the spherical neighborhood approach. This highlights the substantial advantage of using spherical neighborhoods in capturing contextual information and enhancing classification accuracy.

Spherical neighborhoods consider points within a 3D spherical region, which inherently captures shape information of the surrounding objects. This additional information can be beneficial for distinguishing between classes with similar appearance but different shapes. Furthermore, the feature given by the number of points in the neighborhood is useless with the KNN neighborhood definition and do not bring any information about the neighborhood occupancy rate.

### 2.3 Impact of Multiscale Features on Performance

Relying solely on the original scale consistently yields the lowest scores, as demonstrated in Tables 4 and 5. This observation confirms the importance of the multiscale approach. Indeed, using a fixed scale across the scene is inadequate because most scenes contain objects of various sizes. Given the data acquisition process, dense and accurate 3D point clouds exhibiting substantial variations in point density may be expected. This underscores the necessity of considering multiple scales to effectively capture the complexities and variations present in the point cloud.



**Figure 2 :** MDI feature importances with the 18 features at each scale defined in Table 1.  $\mathcal{M}$  refers to the moment.

### 2.4 Comparison with Additional Features and Assessment of Feature Importance

Height features, such as vertical range, as listed in Table 2, were employed in previous studies [2, 4] for their ability to better characterize vertical and thin objects such as traffic signs or tree trunks.

In table 4, it is observed that the IoU scores with or without these features statistically fall within an overlapping range. To further analyze the significance of each feature, their importances were investigated.

Feature importances can be computed as the mean and standard deviation of the accumulation of impurity decrease within each tree. Impurity decrease refers to the reduction in the criterion value<sup>4</sup> when a feature is used to split a node in a decision tree. The importance of a feature is calculated as the (normalized) total reduction of the criterion brought by that feature.

Figures 2 and 3 display the Mean Decrease Impurity (MDI) feature importances and their standard deviation, respectively for the 18 features of the proposed method [5] and the 21 features including the height features, for each scale.

In Figure 2, the feature at the original scale are less important than the feature at smallest scales. Moreover, the "Number of points" gets more important as the scale decreases, validating the interest of using this new feature. Figure 3 highlights that the importances of height features are not negligible, especially at the smallest scale. This highlights the importance of incorporating additional height information into the feature set.

Permutation feature importance is a common method used to assess the significance of features in a classifier prediction, regardless of the model being used. The main idea is to assess how much the model's performance changes when modifying the value of the given feature. This is achieved by randomly shuffling the values of the feature and observing the resulting change in prediction accuracy. Unlike the previous method, this approach is not biased by high-cardinality features, ie variables with a high level of diversity in their values.

Figures 4 and 5 display the permutation feature importances and their standard deviation for the same features.

It is noteworthy that the verticality of the smallest eigenvector  $e_3$  exhibits a significant MDI and permutation feature importance at almost every scales. This observation is consistent with findings in [5], where it was noted that the verticality of the smallest eigenvectors encodes information about the normal vector's orientation of planar objects. This feature's importance stems from its ability to capture the orientation of objects relative to the ground, which can be crucial for distinguishing between different classes, such as 'Motorcycles', 'Pedestrians', and 'Cars'.

## 2.5 NPM3D dataset and generalization on an Unseen point cloud

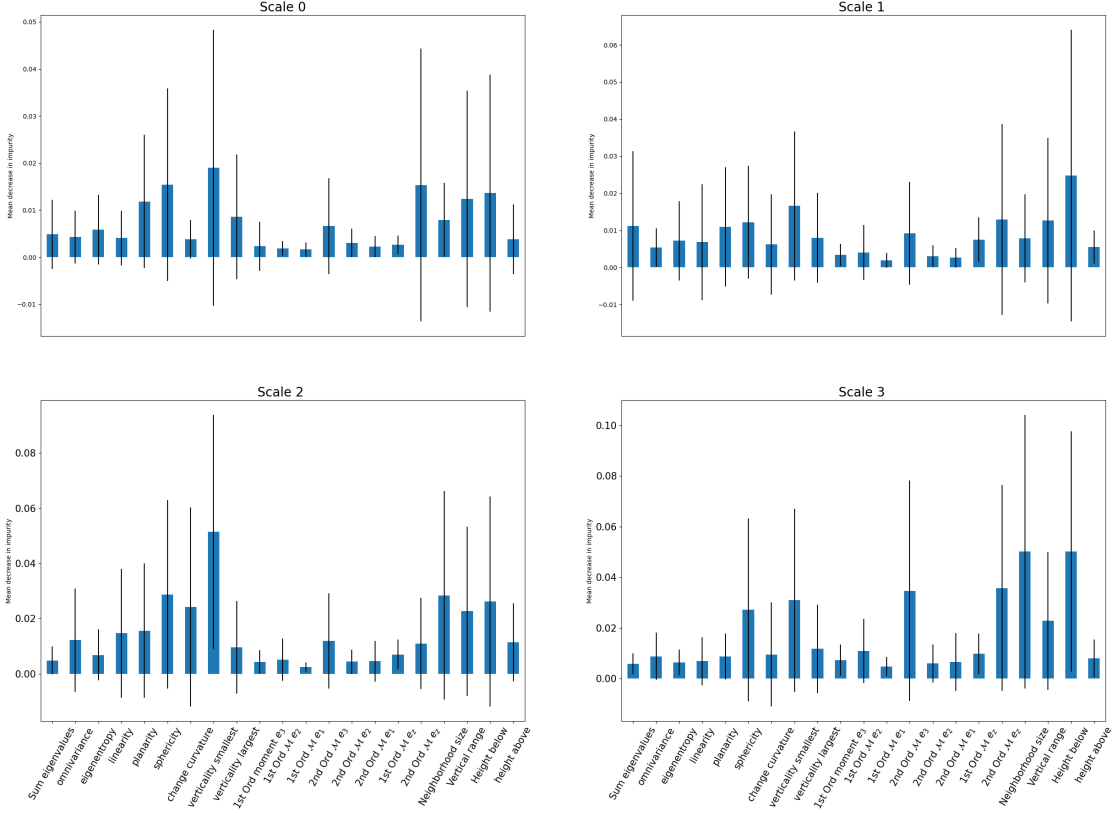
For this experiment, I have randomly selected 100,000 points per class from the MiniLille dataset to train the Random Forest classifier. Here, the number of scales is  $S = 3$ ,  $r_0 = 0.8$ ,  $\phi = 2$ ,  $\rho = 5$  and I used a spherical neighborhood. To assess the generalization ability of the method, I considered points in the MiniParis point cloud with non-zero labels. Subsequently, I predicted the labels of 34,7651 points from this dataset. The results are presented in Table 6.

The classifier trained on the MiniLille dataset generalizes well to the MiniParis dataset, with a weighted IoU score of 97.89%. This demonstrates the robustness of the proposed method and its ability to generalize to this dataset. However, for a fair conclusion, we should also experiment on different large scale datasets.

The computation of features for the test points required 4 hours, while the training of the classifier itself took 49 minutes. These processing times reflect the computational resources necessary for the feature extraction and training phases when dealing with a large number of points.

---

<sup>4</sup>Here, it is the Gini index.



**Figure 3 :** MDI feature importances for the Paris-rue-Cassette dataset with additional height features (21 features in total).

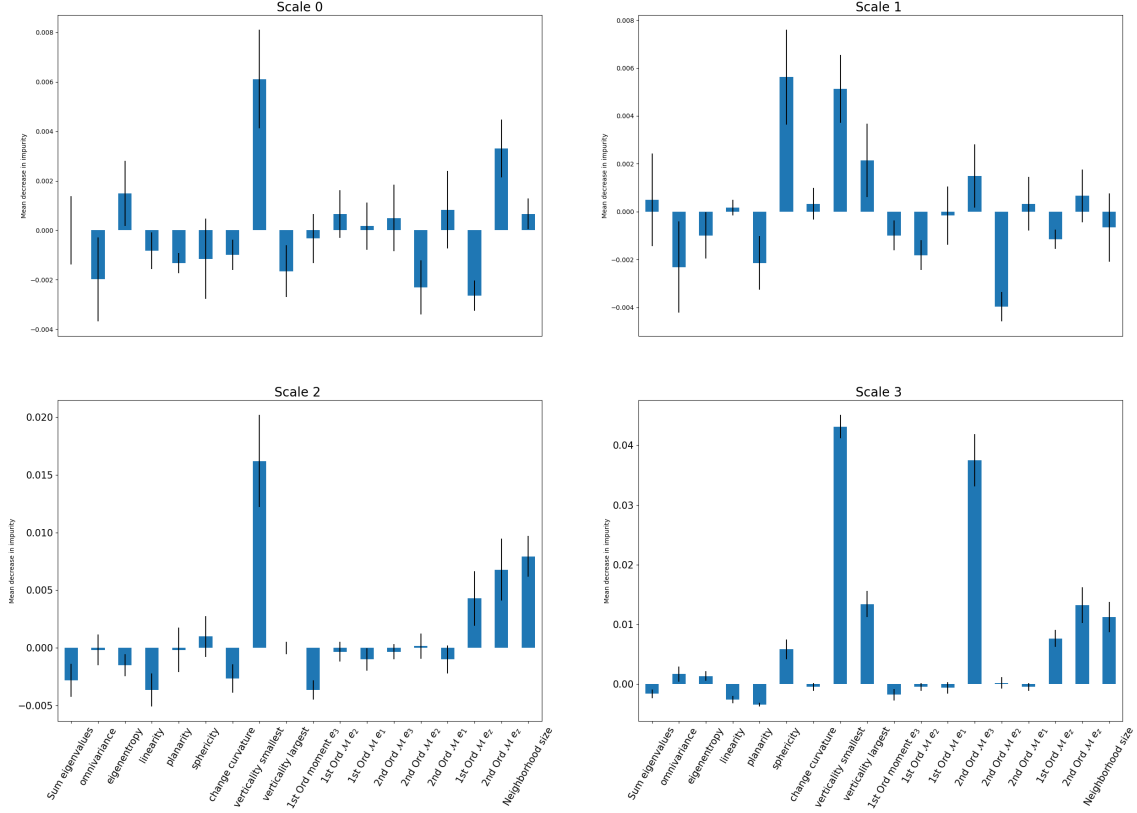
Classes	IoU
Ground	98.41%
Building	97.44%
Poles	98.40%
Pedestrians	98.34%
Cars	98.85%
Vegetation	96.86%
Weighted IoU	97.89%

**Table 6 :** IoU on the MiniParis dataset using the Random Forest classifier trained on the MiniLille dataset. Here, the number of scales is  $S = 3$ ,  $r_0 = 0.8$ ,  $\phi = 2$ ,  $\rho = 5$ .

### 3 Proposals for Improvement

As part of the project’s requirements, proposals for enhancing the method were considered and implemented. The addition of height features to the feature set were discussed in Section 2.4.

**Speed Optimization Attempts.** Efforts were made to accelerate the feature computation process. Initially, I experimented with parallelization using the `multiprocessing` package to distribute computations across multiple CPU processes. However, the overhead introduced by pickling and unpickling large objects transferred over multiprocessing queues proved to be a limiting factor, par-



**Figure 4 :** Permutation feature importances with the 18 features used in [5] at each scale.  $\mathcal{M}$  refers to the moment.

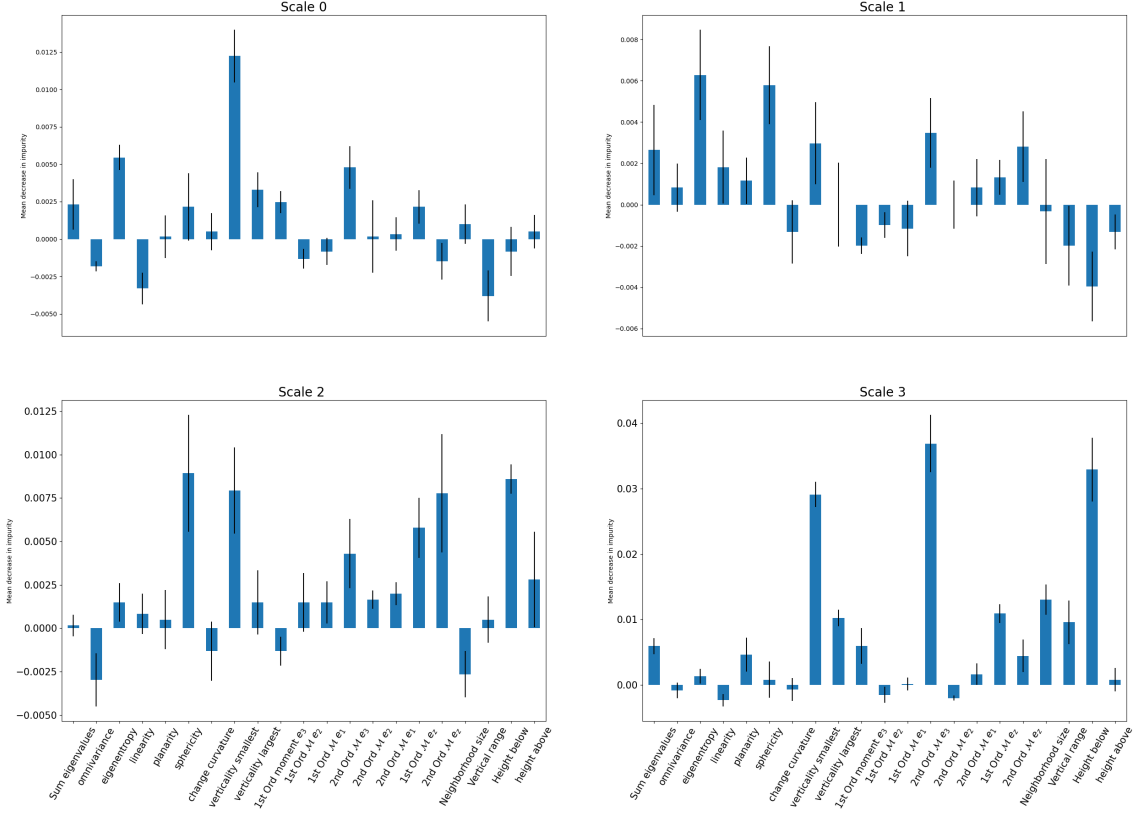
ticularly for datasets with substantial size. This is because large objects require more data to be pickled and transferred, and the unpickling step requires reconstructing the entire object.

Subsequently, I explored the use of Numba, a just-in-time compiler for Python that works best on code that uses NumPy arrays and functions. Computing the features of 40 points with 4 scales on the subsampled Paris-rue-Cassette point cloud (about 1,288,215 points) took 54 seconds. I computed and saved the features of all points, excluding those with label '*Unclassified*'. This process took 5 hours.

**Alternative Classification Method.** Random Forest classifier is commonly used for semantic classification of 3D point clouds [5, 2]. It is well-suited for multi-class problems and has been shown to yield good results in reasonable time on large point clouds [7, 1]. Nevertheless, alternative classification methods were explored in a study by Atik et al. [1], where the authors compared different classifiers across various datasets. These classifiers encompassed Random Forest, a Multilayer Perceptron algorithm, Support Vector Machines, and Linear Discriminant Analysis. The authors observed that the performance of classifiers varied depending on the dataset.

I decided to explore boosting algorithms. Boosting algorithms, such as AdaBoost, and Gradient Boosting are well-known ensemble machine learning approaches that combine multiple weak models to create a robust classifier. I compared the results obtained on the Paris-rue-Cassette point cloud (see Table 4) with those obtained using a Gradient Tree Boosting classifier [3]. Similar to the Random Forest classifier used in Section 2, this approach used 150 trees and the Gini criterion, with the same class weights applied to balance class frequencies. The results are presented in Table 5.





**Figure 5 :** Permutation feature importances for the Paris-rue-Cassette dataset with additional height features (21 features in total for each scale).  $\mathcal{M}$  refers to the moment.

While those results did not yield a significant enhancement, we still observed improvements in overall weighted Intersection over Union (IoU). For the 'Cars' class it enhanced the mean IoU by 22% of the score obtained with the Random Forest classifier. These observations underscored the potential of boosting classifiers to enhance classification performance.

## 4 Conclusion

In this work, I investigated semantic classification using multiscale features and provided valuable insights into the efficiency of various feature extraction approaches and their impact on classification performance. Through comparative analysis, I have demonstrated the superiority of employing spherical neighborhoods over KNN methods, emphasizing the importance of contextual information in accurately characterizing point cloud data.

Additionally, the assessment of additional height features, has underscored their significance in capturing vertical characteristics. Importantly, my findings align with previous research, highlighting the importance of considering spatial orientation in distinguishing between different object classes. Moreover, I have studied the robust generalization of the method to unseen datasets, as evidenced by its performance on the MiniParis dataset using only the MiniLille dataset for training.

Regarding the classification algorithms, while Random Forest Classifier works well, other classifier could be considered. I have evaluated the approach with Gradient Tree Boosting classifier but alternative algorithm could be explored.

Looking ahead, alternative semantic classification methods without the need for explicit feature computation have shown promising results. For instance, In 2019, Thomas et al. [6] proposed a deep learning approach for semantic classification using convolutions on point clouds in the Kernel

Point Convolution (KPConv) network, which operates directly on radius neighborhoods as input. This avenue presents exciting possibilities for further research and development in the field.

## References

- [1] Muhammed Enes Atik, Zaide Duran, and Dursun Zafer Seker. Machine learning-based supervised classification of point clouds using multiscale geometric features. 10(3):187. Number: 3 Publisher: Multidisciplinary Digital Publishing Institute.
- [2] Timo Hackel, Jan D Wegner, and Konrad Schindler. FAST SEMANTIC SEGMENTATION OF 3d POINT CLOUDS WITH STRONGLY VARYING DENSITY.
- [3] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.
- [4] Mahmoud Mohamed, Salem Morsy, and Adel El-Shazly. Improvement of 3d LiDAR point cloud classification of urban road environment based on random forest classifier. 37:1–22.
- [5] Hugues Thomas, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Yann Le Gall. Semantic classification of 3d point clouds with multiscale spherical neighborhoods.
- [6] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Leonidas J. Guibas. KPConv: Flexible and deformable convolution for point clouds.
- [7] Martin Weinmann, Boris Jutzi, Stefan Hinz, and Clément Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. 105:286–304.