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

Abstract

The project consists in studying the article from Thomas et al. [4] by implementing it and proposing improvements. The goal of the studied article is to classify each point of a 3D point clouds using an innovative approach to design features which will be used to train a random forest classifier. [what i did] 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 [4], 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 adating the scale for each point [6]. Indeed, using a fixed scale across the scene is inadequate because most scenes contain objects of various sizes. 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/ρ. 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, 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, ϕ is the ratio between the radius of consecutive neighborhoods and ρ is the ratio between the radius of the spherical neighborhood and the voxel size of the grid subsampling. ρ 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 (×2)	$\lvert \arcsin(\langle e_i, e_z \rangle) \rvert_{i=1,3}$
Absolute moment (×6)	$\frac{1}{\lvert\mathcal{N}\rvert} \lvert \sum_{j\in\mathcal{N}} \langle p_j - \mathbf{p}, e_i \rangle^k \rvert_{k=1,2;\ i=1,2,3}$
Vertical moment (×2)	$\frac{1}{\lvert\mathcal{N}\rvert} \lvert \sum_{j\in\mathcal{N}} \langle p_j - \mathbf{p}, e_z \rangle^k \rvert_{k=1,2}$
Number of points	$\lvert\mathcal{N}\rvert$

Table 1: Geometric features [4] of point $\mathbf{p}$. Here, $N_{feats} = 18$ and λ_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, 3], 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 $\mathbf{p}$.

2 Experiments and results

I used two datasets to test the method: the Paris-rue-Cassette dataset[2], a point cloud of 12 million point and the NPM3D dataset[3], that groups three small point cloud whose 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 NPM3D dataset is used to evaluate how the method generalizes to unseen data. [REF TO THE SECTION]

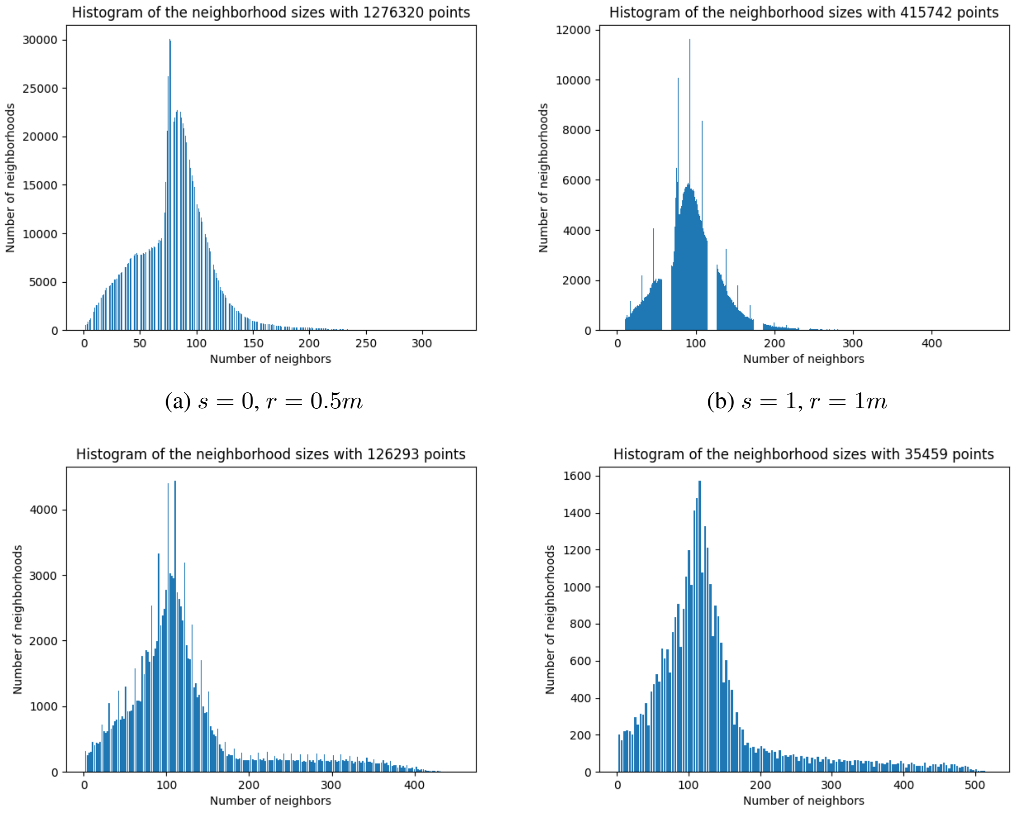
2.1 Paris-rue-cassette dataset

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 the articles [4, 2, 6].

For sake of computation time, I subsampled the point cloud by grid subsampling with a voxel size of 0.1m. The resulting point cloud has 1,288,215 points. The label of each voxel center is the label of the majority of the points in the voxel. Some class has seen their number of points decrease drastically, like the Traffic Sign label. The number of points per class is given in Table 3. It would have been possible to keep all points from those classes on the subsampled point cloud, but it would be interesting to see how we can handle class imbalance in Section 3.3.

[2] http://data.ign.fr/benchmarks/UrbanAnalysis/

[3] https://npm3d.fr/benchmark-for-master-course-on-3d-point-clouds



(a) $s = 0$, $r = 0.5m$

(b) $s = 1$, $r = 1m$

(c) $s = 2$, $r = 2m$

(d) $s = 3$, $r = 4m$

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.

Label	Size
Ground	188644
Building	954267
Traffic Signs	195
Pedestrians	2079
Cars	28437
Vegetation	98175
Motorcycles	4523

Table 3: Class size in Cassette subsampled cloud

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

2.2 Influence of neighborhood definition

With spherical neighborhoods instead of KNN, the features always describe a part of the space of the same size at each scale.

Method	Ground	Building	Traffic Signs	Pedestrians	Cars	Vegetation	Motorcycles	Weighted IoU
[4]	81.48 ± 2.95	0.71 ± 0.08	0.90 ± 0.11	0.93 ± 0.02	0.47 ± 0.13	0.82 ± 0.05	0.97 ± 0.02	0.79 ± 0.04
W height features	83.16 ± 2.23	0.72 ± 0.04	0.97 ± 0.05	0.94 ± 0.01	0.50 ± 0.13	0.81 ± 0.06	0.95 ± 0.03	0.79 ± 0.03
W/O multi-scaling	68.76 ± 5.77	0.43 ± 0.03	0.09 ± 0.03	0.71 ± 0.02	0.30 ± 0.04	0.63 ± 0.03	0.55 ± 0.03	0.55 ± 0.02
Knn neighborhood	71.87 ± 8.28	0.57 ± 0.06	0.48 ± 0.27	0.91 ± 0.02	0.37 ± 0.08	0.81 ± 0.02	0.87 ± 0.02	0.71 ± 0.03

Table 4: Benchmark results with Random Forest classifier

Method	Ground	Building	Traffic Signs	Pedestrians	Cars	Vegetation	Motorcycles	Weighted IoU
[4]	83.34 ± 2.82	0.70 ± 0.06	0.59 ± 0.25	0.93 ± 0.02	0.61 ± 0.10	0.78 ± 0.06	0.96 ± 0.03	0.80 ± 0.03
W height features	84.94 ± 2.38	0.74 ± 0.07	0.82 ± 0.18	0.94 ± 0.02	0.59 ± 0.12	0.79 ± 0.06	0.98 ± 0.01	0.82 ± 0.04
W/O multi-scaling	62.13 ± 8.48	0.41 ± 0.04	0.09 ± 0.01	0.71 ± 0.03	0.29 ± 0.04	0.67 ± 0.02	0.60 ± 0.03	0.55 ± 0.02
Knn neighborhood	71.06 ± 9.34	0.56 ± 0.09	0.37 ± 0.26	0.94 ± 0.02	0.50 ± 0.14	0.81 ± 0.04	0.91 ± 0.03	0.74 ± 0.03

Table 5: Benchmark results with HBG classifier

2.3 Influence of scales

2.4 Additional features and feature importance comparision

Feature importances are computed as the mean and standard deviation of accumulation of the impurity decrease within each tree. Impurity decrease is the reduction of the criterion value (such as entropy or gini index) when a feature is used to split a node in a decision tree. The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature.

Permutation feature importance is a model agnostic feature importance mehtod and is a commonly used technique to assess how much a feature is important to a classifier prediction. The idea is to assess how much the model's performance changes when modifying the value of the given feature. It is done by shuffling the values of the variable and observe the difference of the prediction accuracy. Unlike the previous method, this approach is not biased by high-cardinality features.

With a uniform density, the number of points in the neighborhood becomes a feature itself, describing the neighborhood occupancy rate. In Figure 2, the number of points in the neighborhood is one the most important features, especially at the smallest scale.

2.5 NPM3D dataset

3 Improvement proposals

3.1 Speed up attempts

I tried to compute the features on multiple CPU processes using `multiprocessing` package to speed for loops. However, objects placed on multiprocessing queues are pickled, transferred over the queue, and then unpickled. The pickling and unpickling steps add overhead, and for large objects this overhead can be significant. This is because large objects require more data to be pickled and transferred, and the unpickling step requires reconstructing the entire object.

Then, I tried Numba which is 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 points (about 1,288,215 points) took 54 seconds. I computed and saved the features of all points, excluding the points with label 0. It took in [time] hours.

3.2 Active learning

3.3 Classification methods

Random Forest classifier is usually used for semantic classification of 3D point clouds [4, 2]. Indeed, it is directly applicable to multi-class problems and has been shown to yield good results in reasonable time on large point clouds [6, 1]. I use Gini index as splitting criterion.

However, other classification methods could be considered [1].

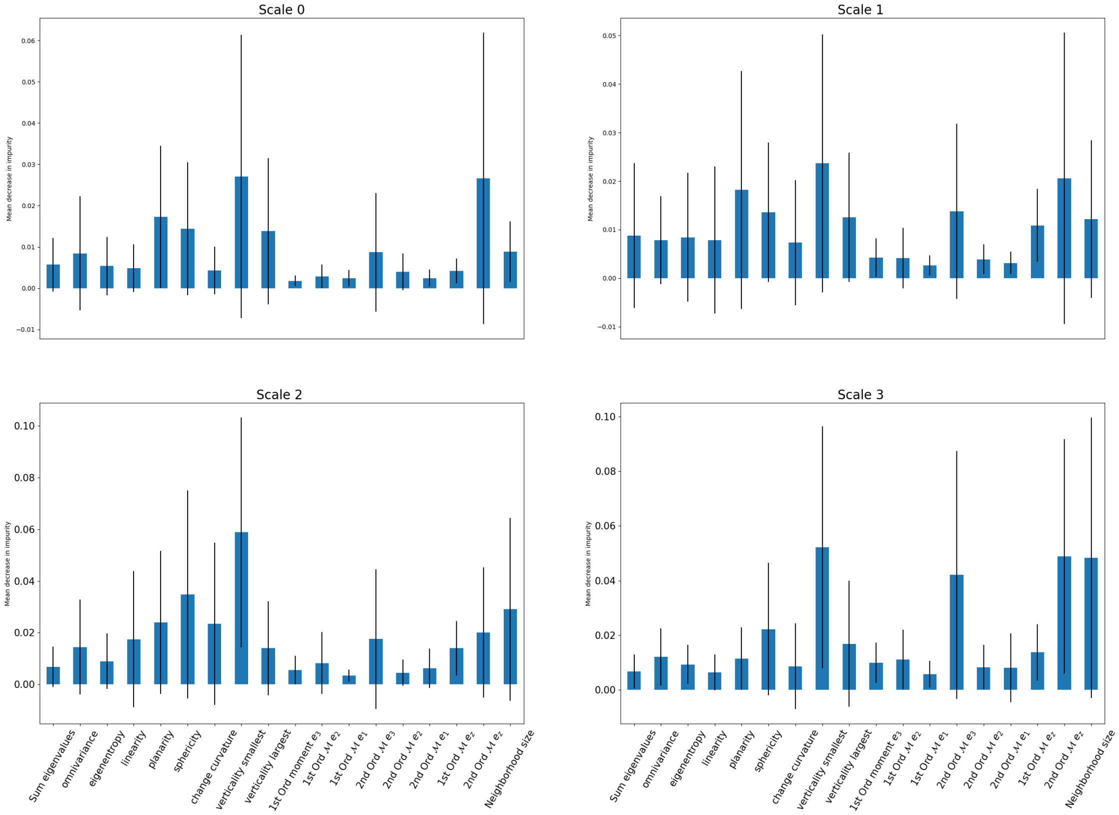


Figure 2: Mean decrease impurity feature importances with the 18 default features

Boosting is an ensemble machine learning approach. Boosting algorithms combine multiple low accuracy or weak models to create high accuracy or strong model. Among the most popular boosting algorithms are AdaBoost, Gradient Boosting, and XGBoost.

4 Conclusion

In 2019, Thomas et al. [5] propose a deep learning approach for semantic segmentation using convolutions on point clouds.

References

[1] Muhammed Enes Atik, Zaide Duran, and Dursun Zafer Seker. Machine learning-based supervised classification of point clouds using multiscale geometric features. *ISPRS International Journal of Geo-Information*, 10(3), 2021.

[2] T. Hackel, J. D. Wegner, and K. Schindler. Fast semantic segmentation of 3d point clouds with strongly varying density. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, III-3:177–184, 2016.

[3] 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.

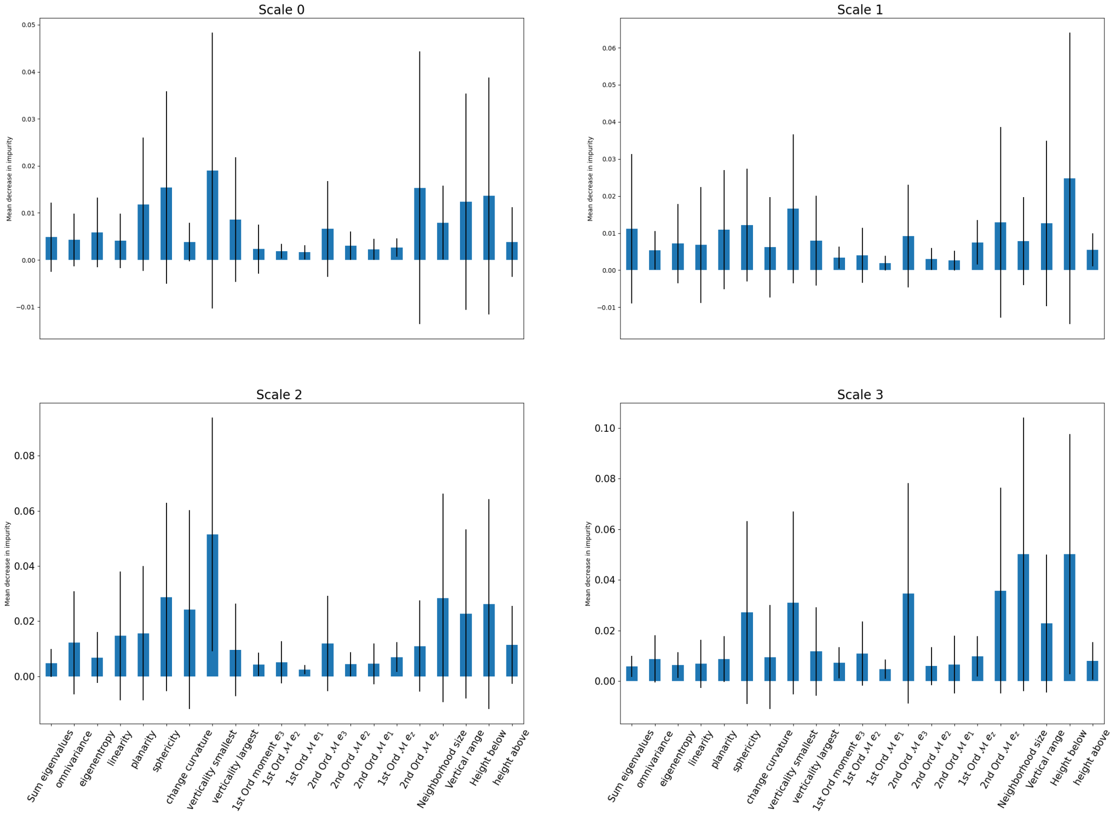


Figure 3: Mean decrease impurity feature importances for the Paris-rue-Cassette dataset with additional height features (21 features in total).

[4] Hugues Thomas, Jean-Emmanuel Deschaud, Beatriz Marcotegui, François Goulette, and Yann Le Gall. Semantic classification of 3d point clouds with multiscale spherical neighborhoods.

[5] Hugues Thomas, Charles R. Qi, Jean-Emmanuel Deschaud, Beatriz Marcotegui, FranÃ§ois Goulette, and Leonidas J. Guibas. KPConv: Flexible and deformable convolution for point clouds.

[6] Martin Weinmann, Boris Jutzi, Stefan Hinz, and Clément Mallet. Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105:286–304.

A Multiscale features computation

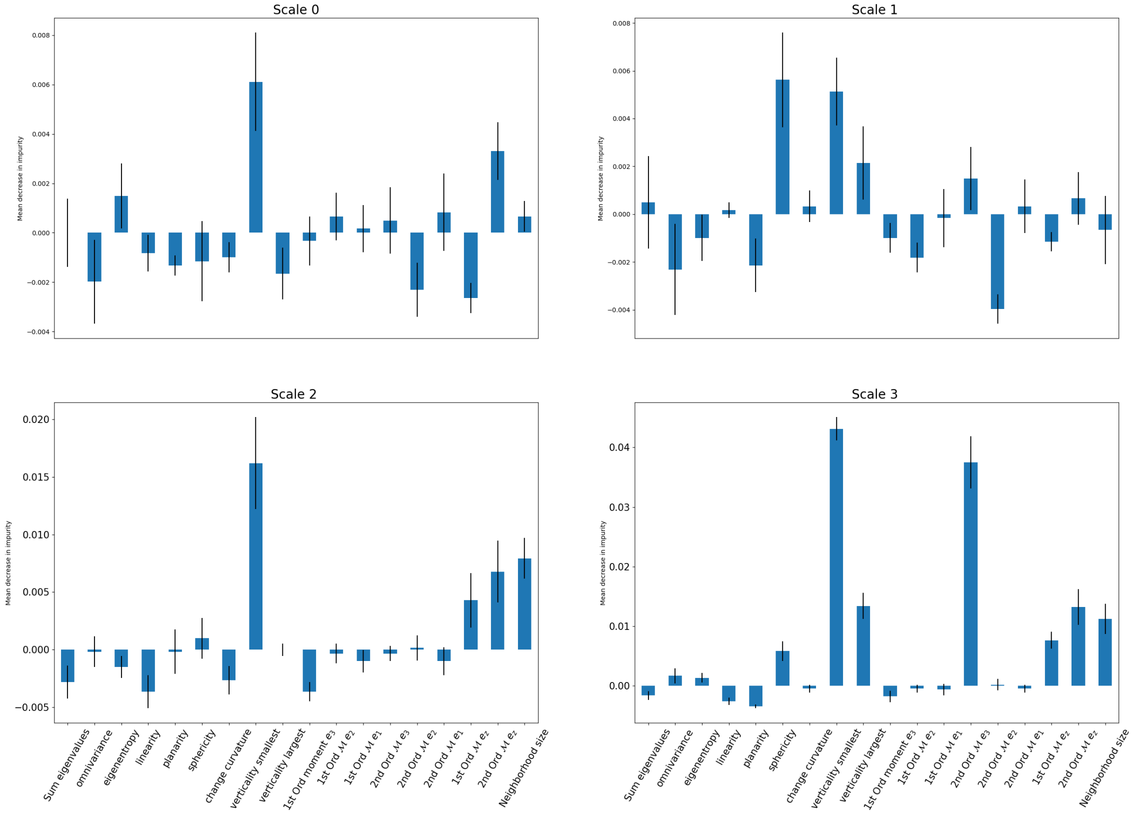


Figure 4: Permutation feature importances with th default 18 features

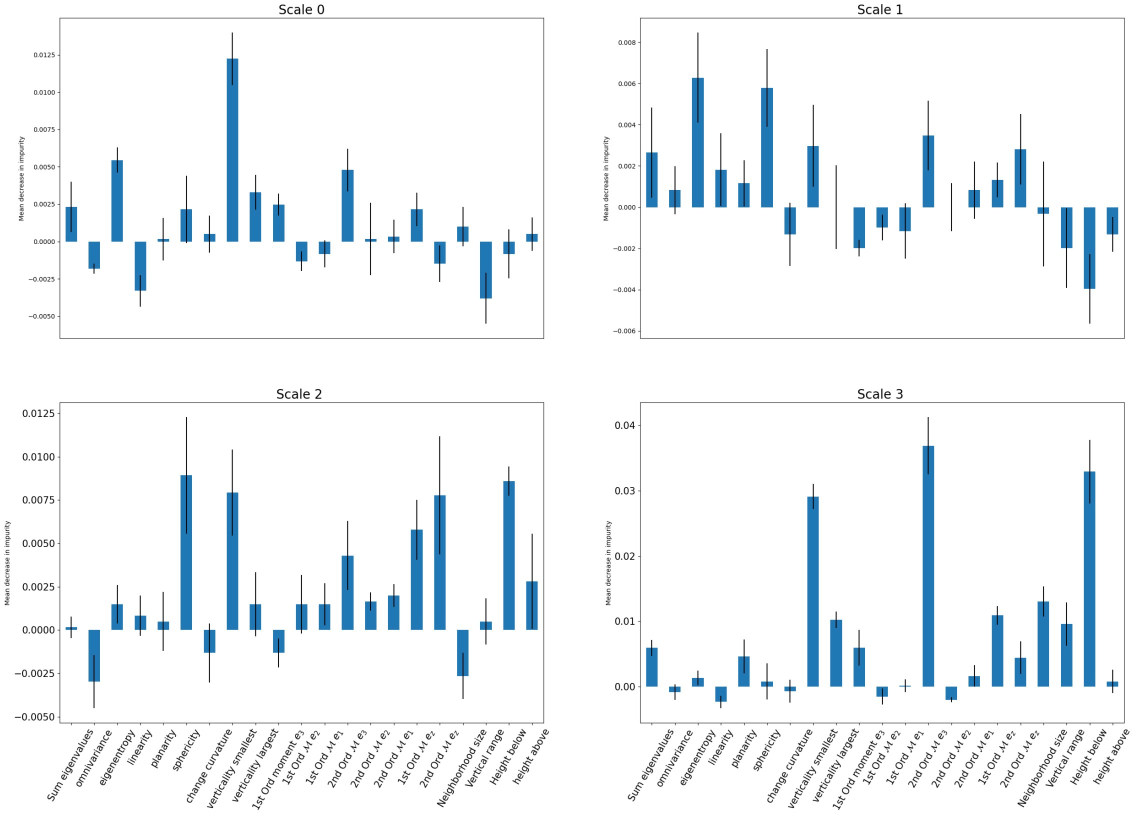


Figure 5: Permutation feature importances for the Paris-rue-Cassette dataset with addition height features (21 features in total).