IADIS International Journal on Computer Science and Information Systems
Vol. 20, No. 2, pp. 82-95
ISSN: 1646-3692

AUTOMATED MACHINE LEARNING FOR HYPERPARAMETER OPTIMIZATION IN POINT CLOUD PART SEGMENTATION

Gabriel Lenz Balatka & Rafael Stubs Parpinelli
Graduate Program in Applied Computing - Santa Catarina State University – Joinville/SC, Brazil

ABSTRACT

The point cloud part segmentation task consists of segmenting an object, represented by a point cloud, into its constituent parts, such as a chair that is segmented into seat, backrest, and legs. The most recent computational strategies use Artificial Neural Networks to perform this task, but the architectures used are developed generically and therefore do not consider the specific patterns of each category of objects. Thus, this work proposes to analyze the contribution of building specific architectures based on the optimization of hyperparameters of the PointNet architecture, which is well established in the literature. The dataset used was the PartNet, and four case studies were employed. In addition, we also studied the impact of point cloud size on this segmentation task, performing the optimization process in each category studied in three different point cloud sizes: 512, 1,024, and 2,048. From the results obtained, an average improvement of 2% in the test accuracy metric was achieved in the Table-1, Chair-1, and Lamp-1 categories and 6% in the StorageFurniture-1 category. The impact of point cloud size was low, and statistically significant improvements were observed in Table-1, Chair-1, and StorageFurniture-1 categories. Thus, hyperparameter optimization proved to be consistent, achieving satisfactory results.

KEYWORDS

Point Cloud Subsampling, Point Cloud Part Segmentation, Automated Machine Learning, Artificial Neural Network, Hyperparameter Tuning

1. INTRODUCTION

Point clouds are three-dimensional sets of points that can be used to represent objects and physical spaces. By processing such structures, it is possible to automate different processes, mainly related to industry. In this work, three different terms will be used frequently: category, shape, and part. Category is a generic type of object, for example, a table or a chair. Shape is a single object that belongs to a category, represented by a point cloud. In addition, categories have parts, for example, in the case of a chair, we can divide it into arms, legs, seat, and backrest.

Note that a shape does not necessarily have points for all parts, as in the case of a chair without arms (Qi, 2017).

This paper studies the application of a process called Point Cloud Part Segmentation. To do this, we start with a point cloud whose category is known. This application aims to use a model-specific trained for this category that receives this point cloud and returns a label for each point, i.e., the part of the shape that it belongs to. Thus, starting from a known and well-established neural network architecture for this application, called PointNet (Qi et al., 2017), this work aims to carry out an optimization process of the hyperparameters of the original architecture, making a comparison between the results obtained.

The PointNet artificial neural network architecture requires data for training and testing. However, each set of data has a specific behavior, i.e., the shapes have specific patterns, and the number of parts in each category is also different. The goal of this research is to study the importance of hyperparameter tuning of a generic architecture to build specific ones that are more suitable for each case. The dataset used was PartNet (Mo et al., 2019), available online at https://huggingface.co. The PartNet dataset is a subset of the ShapeNet dataset (Chang et al., 2015) and consists of a total of 26,671 distinct shape instances, separated into 24 categories.

2. BACKGROUND

2.1 Point Clouds

Point Cloud is an important computational structure used in the field of Geometric Modeling. Using specific sensors, it is possible to extract this information to represent environments as a whole or individual object. Thus, according to (Wang & Kim, 2019), point cloud processing is currently increasingly sought after to extract relevant information from environments or objects, mainly for industrial automation. Another widely explored application, as in the article by (Zermas Izzat & Papanikolopoulos, 2017), is autonomous vehicles, by extracting the environment with a LiDAR sensor and being able to recognize objects close to the vehicle.

From a Point Cloud, many computational processes can be carried out (Qi et al., 2017). When this cloud represents an unknown three-dimensional shape, the aim is to classify it from a list of possible categories. The goal is to find the category that best corresponds to the respective point cloud. On the other hand, when the category that the cloud represents is already known, assuming, for example, a chair, and starting with which parts represent this category, in this case, the arms, legs, seat and backrest, the goal is to try to find which part of the shape each point in the cloud belongs to. This process is known as point cloud part segmentation.

Currently, the most widely used solution to achieve this process is the development of an Artificial Neural Network, a strategy initially proposed by (Qi, 2017), in which it depends on one main process, without any intermediate processing to convert the initial structure type. This generally gives better results with less processing time. In this work, the segmentation application will be explored using Artificial Neural Networks. Firstly, we will evaluate the performance of a specific architecture called PointNet (Qi, 2017). Secondly, Automated machine learning will then be utilized to adjust the hyperparameters of the respective architecture, ending with a new performance evaluation.

2.2 Automated Machine Learning

Automated Machine Learning (AutoML) is characterized by the automation of the entire Machine Learning process, from data preparation to model evaluation. Different objectives are assigned to this process, such as the construction of complex systems without the need for extensive knowledge in Statistics and Machine Learning (He, Zhao & Chu, 2021).

The AutoML process is made up of a series of stages, namely data processing, feature engineering, model/algorithm selection, and hyperparameter optimization (Salehin, 2024). Hyperparameter optimization seeks to optimize parameters related to training, such as learning rate and batch size (Salehin, 2024). The objectives of hyperparameter optimization are to reduce human effort, improve algorithm performance, and increase the reproducibility of scientific studies (Hutter, Kotthoff & Vanschoren, 2019). On the other hand, there are challenges such as high computational costs, complex search space, and limited data sets (Hutter et al., 2019).

Hyperparameter Optimization starts with a known configuration, previously found manually or automatically, or from a random configuration, and throughout the iterations of the selected optimization algorithm, new configurations are tested and evaluated according to the chosen metric. The configurations are within a search space defined by the programmer. Different optimization algorithms can be used for this, such as Grid Search, Random Search, Bayesian Optimization, and HyperBand (Hutter et al., 2019). The Random Search algorithm, for example, tests the initial configuration and then selects random configurations from the search space, without any kind of strategy. In this work, we opted for Bayesian Optimization, a method widely used in the optimization of functions with a high computational cost of evaluation (Wang et al., 2023). In addition, the Gaussian Process was used as the surrogate model in Bayesian Optimization, the most widely used model in this optimization technique (Wang et al., 2023).

3. RELATED WORK

In the work of (Daif & Marzouk, 2025), the PointNet architecture was used to classify and segment point clouds of steel structure elements. Classification took place between two categories and segmentation into five parts. Optimization occurred manually by adjusting the number of epochs, number of points per cloud, and batch size. Due to the lower complexity of the problem, no AutoML technique was needed to optimize the hyperparameters, and 100% accuracy was achieved for classification and 97.10% accuracy for part segmentation.

In (Wielgosz et al., 2023), a framework was proposed for semantic and instance segmentation for coniferous forests, on two different datasets. At the end, Bayesian Optimization was applied to tune hyperparameters, through 50 total iterations. A total of six hyperparameters were optimized, based on a search space defined by the authors themselves. The overall performance of the framework improved by around 4% for the F1 score metric. The study of hyperparameter tuning has proved important in literature in general.

No article was found that it carried out the same tuning process for point cloud part segmentation with the PartNet dataset. This opens a new avenue for studying this subject. Furthermore, in the article by (Qi et al., 2017), in addition to proposing the PointNet architecture, a robustness test was performed, but on the task of point cloud classification. For this, the authors used the ModelNet40 dataset, proposed by (Wu et al., 2015), and verified the performance of the PointNet architecture in reducing the size of point clouds. As a result, the authors observed a slight reduction in the accuracy metric. However, by significantly subsampling the point clouds, the authors noticed a drastic reduction in the accuracy metric. Further studies can be conducted on the impact of point cloud subsampling on the processing tasks of this data type, as well as studying this impact specifically on the part segmentation task.

4. PROPOSED APPROACH

Based on the discussion in Section 3, this work aims to analyze the contribution of hyperparameter optimization in Artificial Neural Networks (ANNs) for the Point Cloud Part Segmentation task. The different ANN architectures developed for this task achieve promising results in different categories. However, since each category has its own pattern, creating a new optimized architecture for each one can significantly improve performance. In addition, considering that Artificial Neural Networks work with fixed-size point clouds, it is also important to compare the results achieved in different point cloud sizes.

For training and testing the models, we used the PartNet dataset, published by (Mo et al., 2019), due to its wide variety of data, which contains a total of 26,671 distinct shape instances divided into 24 categories. For this purpose, we selected four categories from this dataset: Table, Chair, Storage Furniture, and Lamp. The PartNet dataset provides three levels of detail for each category, ranging from coarse to fine-grained. We used the coarse degree of detail, which allows for a better qualitative analysis of the results, as the shapes are composed of fewer parts compared to other levels. In addition, we followed the dataset notation, adding suffix 1 to each category to indicate that the coarse level of detail was utilized. These categories were selected because they contain the largest number of shapes available, thus achieving a fairer comparison.

To perform the experiments, for each category, the number of points of each shape will be subsampled to 512, 1,024, and 2,048 points. In the PartNet dataset, all available point clouds have a size of 10,000 points, and the sizes defined for simplifying these clouds were chosen empirically, according to the amount of processing available and the representativeness of the objects, considering that very small point clouds, for example, with only 100 points, may not be sufficient to represent complex objects.

In this process, each combination of category and number of points per shape will be evaluated by comparing two architectures: Standard and Best. The Standard architecture refers to the PointNet developed by (Qi et al., 2017). For the Best architecture, hyperparameter optimization will be conducted on the Standard architecture for each case. Finally, both architectures will be trained and tested 10 times using data exclusively from the respective category under analysis.

For training and testing Artificial Neural Networks, it is necessary to divide the dataset into training, validation, and testing sets. The PartNet dataset already separates the instances into the respective sets, and therefore, we chose to perform all experiments following this division. Therefore, the redistribution of the dataset was not performed to avoid losing the possible balance achieved originally. Table 1 shows the split found in each case.

Table 1. The proportion between each split for each category

Category	Train	Validation	Test
Table-1	5,707 (69.45%)	843 (10.26%)	1,668 (20.3%)
Chair-1	4,489 (70.99%)	617 (9.76%)	1,217 (19.25%)
StorageFurniture-1	1,588 (69.99%)	230 (10.14%)	451 (19.88%)
Lamp-1	1,554 (70.41%)	234 (10.60%)	419 (18.99%)

The categories mentioned above have 12, 6, 7, and 18 parts, respectively. Based on this choice, the part segmentation process can be studied at different levels of complexity, according to each shape structure, the form and proportion of the division between the splits, and the number of parts of each category. Moreover, the PartNet dataset is split with approximately 70% for training, 10% for validation, and 20% for testing, as reported by the authors of the dataset.

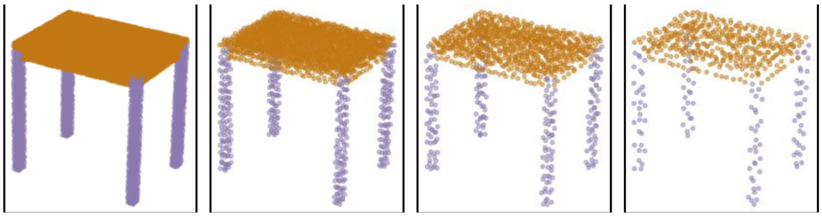
The proportions found between the train, validation, and test sets follow the standard within the field of machine learning. In addition, there was a large difference in the total amount of data available in each of the categories. The Table-1 category has the largest number of shapes available, with a total of 8,218 shapes, followed in descending order by the Chair-1 category with 6,323, StorageFurniture-1 with 2,269, and Lamp-1 with 2,207 shapes. Thus, there is a significant difference in complexity between the Table-1 category, which has many instances available and 12 parts in total, and Lamp-1, which has fewer instances and 18 parts in total. For this work, these differences are very positive, as they allow the analysis of the contribution of hyperparameter optimization in different scenarios.

In this work, two different analyses will be performed to compare the results. The quantitative analyses will examine loss and accuracy metrics using mean and standard deviation, boxplots, and critical difference diagrams. For the qualitative analysis, a comparison will be made of the Ground Truth point clouds, which represent the expected results, and the results achieved by the Standard and Best architectures.

Quantitative analysis, mainly using critical difference diagrams, allows to measure and verify the results obtained statistically, while qualitative analysis, through the analysis of point clouds, enables the understanding and interpretation of results, generating valuable insights for this area of research. In this way, there will be a discussion of the impact of hyperparameter optimization on the point cloud part segmentation task through both analyses performed.

5. PROTOCOL OF EXPERIMENTS

The first process performed was to reduce the size of the point clouds to 512, 1,024, and 2,048 points. For this task, we used the Farthest Point Sampling (FPS) algorithm, which was also used by (Mo et al., 2019) in the construction of the PartNet dataset. The FPS algorithm reduces the number of points in a point cloud, trying to maintain the original format without losing relevant information. Figure 1 shows the results obtained by the algorithm for a specific shape of the Table-1 category, comparing the subsamplings with the original shape of 10,000 points.



(a) Ground Truth (b) 2,048 Points (c) 1,024 Points (d) 512 Points

Figure 1. Application of FPS algorithm to reduce #Points per cloud

By applying the FPS algorithm to the four categories, reducing the point clouds to 512, 1,024, and 2,048 points per cloud, a total of 12 datasets were obtained, following the same split presented in Table 1. Hence, the different datasets of each category differ only in the number of points per cloud. Analyzing the results in Figure 1, the differences between the sizes of the point clouds are clearly visible, but all shapes consistently represent the object of interest. Therefore, the FPS algorithm worked as expected, maintaining the original shape and without losing relevant details of the object.

Moreover, the one-hot encoding technique was utilized to send the appropriate data formatting to the neural networks. Subsequently, the hyperparameters to be tuned were chosen, as well as the search interval for each one. Hence, Table 2 shows the name of the hyperparameters selected from the standard architecture, the default values utilized, and the list of options for each one. The choice of hyperparameters was made empirically, seeking to choose the most significant ones. Thus, convolutional layer filters and hyperparameters related to the learning rate were selected, which significantly impacted the performance of machine learning models in general.

Table 2. List of Options for each hyperparameter selected for tuning

Hyperparameter	Default	List of Options
Features_64	64	16, 32, 64, 128, 256
Feature_128_1	128	32, 64, 128, 256, 512
Features_512	512	128, 256, 512, 1,024, 2,048
Pre_maxpool_block	2,048	512, 1,024, 2,048
Segmentation_features	128	32, 64, 128, 256, 512
Initial_learning_rate	0.003	0.001, 0.002, 0.003, 0.004, 0.005
Decay_steps_multiplier	5	3, 4, 5, 6, 7
Decay_rate	0.5	0.3, 0.4, 0.5, 0.6, 0.7
Staircase	True	False, True

In the list of options of each hyperparameter selected, the values used in the Standard architecture were included as a strategy to ensure that the Best architectures are, in the worst case, equivalent to the Standard architecture. In addition, different strategies were used to choose additional values. First, for the filters in the convolutional layers, we sought to select values proportional to the default ones. For the Boolean hyperparameter, we considered the two possibilities: False and True. Finally, for the other hyperparameters, we added options close to the default ones. All hyperparameters have a total of five options, except for the Boolean one, which has two total possibilities, and the convolutional filter 'Pre_maxpool_block', which has a high initial value, and we preferred to limit this search range to avoid options that were too different from the default value.

After the preprocessing stage, the PointNet architecture was built, using the implementation available at keras.io. The KerasTuner library was used to build the optimization process. The optimization algorithm chosen was the Bayesian Optimizer with 20 trials. In each trial, the algorithm generated a new architecture and evaluated it according to the validation loss metric. Each architecture was trained for a maximum of 100 epochs, and the Early Stopping technique was used with a patience of 10, thus avoiding the processing of unpromising architectures. In addition, the first iteration of the optimization algorithm was configured to use the default hyperparameters, following the strategy of ensuring that the Best architectures were at least equivalent to the Standard one.

Once the best hyperparameters had been found for each study, through the optimization algorithm, both the standard and best architectures were trained and tested a total of 10 times. Each training session consisted of exactly 100 epochs, but after training, the network weights were replaced by the weights found in the epoch with the lowest validation loss. Test accuracy and loss metrics were calculated for each run and will be presented and discussed in Section 6, as well as the comparison between the point clouds generated with both architectures and respective ground truths. Finally, it should also be noted that for preprocessing, training, and testing, a computer with an AMD Ryzen 9 7900X 12-Core processor, RTX 5000 Ada Generation GPU, and 96GB of RAM was used.

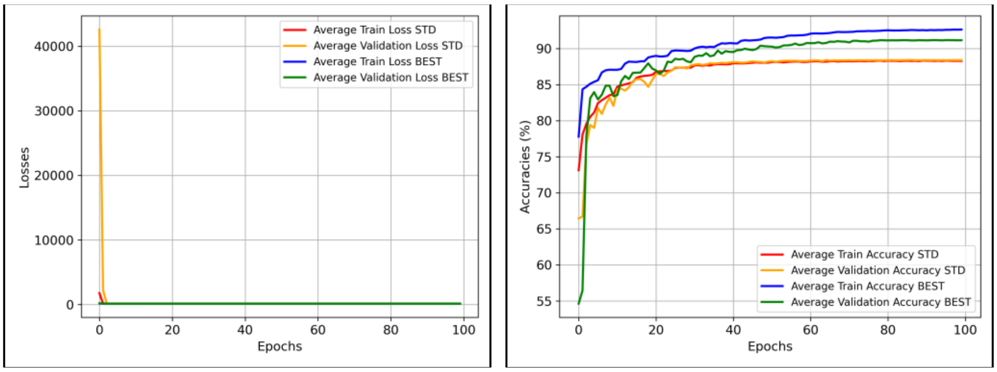
6. ANALYSIS OF RESULTS OBTAINED

Based on the experiments presented in Section 5, this section will present and discuss the results obtained. To this end, this section is divided into two parts: Quantitative and Qualitative Analysis. For the quantitative analysis, tables, boxplots, and critical difference diagrams will be used to analyze the performance of the architectures, and for the qualitative analysis, the objects segmented by the Standard and Best architectures will be compared to the Ground Truths.

6.1 Quantitative Analysis

First, for each combination of category and point cloud size, the convergence of loss and accuracy metrics was analyzed in the training and validation sets in both Standard and Best architectures. For the accuracy metric, adequate convergence was observed in all cases in the training and validation sets. Also, in all cases, the training and validation accuracy in the Best architecture was higher than in the Standard one during the final training epochs. Furthermore, for both architectures, in the training and validation sets, the loss metric convergence was adequate. However, unlike accuracy, the four values were very close to each other, being not possible to verify the impact of the Best architectures in each of the studies performed through the convergence graph.

As the behavior of the convergence graphs was very similar to each other, Figure 2 shows only the convergence graphs of the loss and accuracy metrics for the Table-1 category with a size of 2,048 points per cloud. Both graphs show the convergence of the metrics in the two architectures analyzed during the 100 epochs, represented by indices 0 to 99. In the first epoch of the Standard architecture, the loss in the validation set reached a very high value. This behavior was observed in all Standard architectures, but the values decreased significantly in the following training epochs, becoming close to the other loss metrics.



(a) Mean Loss (b) Mean Accuracy

Figure 2. Loss and Accuracy convergence in both train and validation for Table-1 category with 2,048 points per cloud

Thus, through convergence graphs, very promising results were observed, with Best architectures achieving a higher success rate. However, to verify these results, it is essential to compare the metrics obtained in the test set, composed of point clouds that were not used during the training process in any of the models. To this end, Table 3 presents the loss metric in the test set for the four categories with different point cloud sizes to compare the Standard and Best architectures.

Table 3. Test Loss results with standard and best hyperparameters

#Points	Standard Table-1	Standard Chair-1	Standard StorageFurniture-1	Standard Lamp-1
512	131.47 ± 0.05	131.88 ± 0.15	135.10 ± 2.08	134.47 ± 1.43
1,024	131.46 ± 0.03	132.03 ± 0.31	134.57 ± 1.31	133.43 ± 0.57
2,048	131.48 ± 0.06	131.98 ± 0.18	135.31 ± 2.98	133.55 ± 0.62
#Points	**Best Table-1**	**Best Chair-1**	**Best StorageFurniture-1**	**Best Lamp-1**
512	131.31 ± 0.01	131.58 ± 0.01	131.14 ± 0.03	131.49 ± 0.05
1,024	131.35 ± 0.01	131.53 ± 0.01	131.20 ± 0.02	131.44 ± 0.02
2,048	131.32 ± 0.01	131.53 ± 0.01	131.20 ± 0.02	131.54 ± 0.03

In the Standard architecture, the values were close to each other, ranging from 131 to 136. However, despite being close, even considering the mean and standard deviation, differences are observed. In the Table-1 and Chair-1 categories, which have the largest number of instances available, no significant differences were observed comparing the different sizes of point clouds. Therefore, in these cases, the process of simplifying the point clouds did not alter the results obtained in this metric. In the StorageFurniture-1 and Lamp-1 categories, which have fewer instances available, the simplification of the point clouds also did not significantly affect the results, but greater losses were observed, ranging from 133 to 136.

In the Best architectures, the first thing to note is the standard deviation of all cases, which were low and lower than the Standard architecture. Even so, slightly higher standard deviations were observed in the categories with fewer instances. In relation to the average, all values were close to each other, ranging between 131 and 132. Hence, clear improvements achieved by the Best architectures can be seen in comparison to the Standard one, especially in the StorageFurniture-1 and Lamp-1 categories, which have a smaller number of instances.

Also, it is essential to analyze the accuracy metric in the test set, which measures the number of correct answers, that is, the number of correct points in the respective point clouds. For this purpose, Table 4 presents a comparison of the Standard and Best architectures in the four categories with the three different point cloud sizes.

Table 4. Test Accuracy results with standard and best hyperparameters

#Points	**Standard Table-1**	**Standard Chair-1**	**Standard StorageFurniture-1**	**Standard Lamp-1**
512	88.73% ± 0.84%	87.33% ± 0.97%	71.60% ± 1.51%	67.41% ± 1.93%
1,024	88.61% ± 0.69%	86.83% ± 0.82%	70.80% ± 1.56%	68.27% ± 1.08%
2,048	88.29% ± 0.70%	87.35% ± 1.10%	68.76% ± 2.32%	68.02% ± 0.93%
#Points	**Best Table-1**	**Best Chair-1**	**Best StorageFurniture-1**	**Best Lamp-1**
512	91.33% ± 0.17%	88.88% ± 0.18%	78.41% ± 0.59%	70.66% ± 1.30%
1,024	90.00% ± 0.37%	89.79% ± 0.14%	76.49% ± 0.65%	69.59% ± 0.54%
2,048	90.99% ± 0.22%	89.86% ± 0.23%	75.76% ± 0.49%	70.60% ± 1.13%

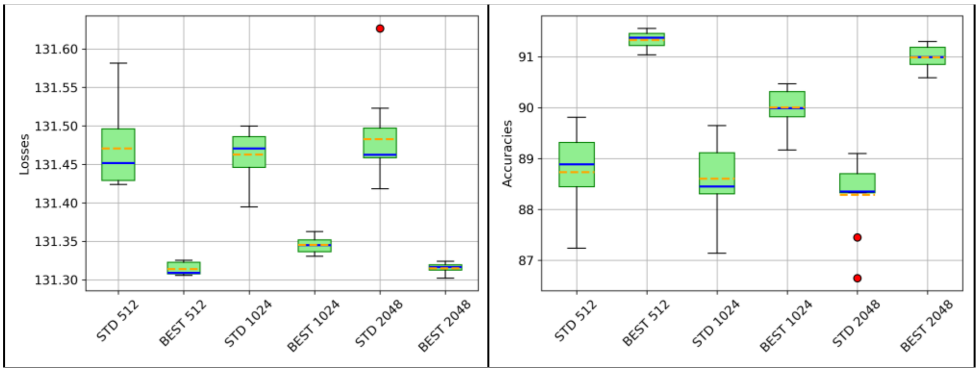
First, in relation to Standard architecture, the impact of the number of instances used during model training is clear. The categories with the highest number of instances achieved accuracies above 85%, while the categories with fewer instances achieved accuracies close to 70%. Regarding the standard deviation, values lower than 3% are observed, with the Chair-1 category obtaining the lowest values, being less than 1% in the three point cloud sizes. On the other hand, the StorageFurniture-1 category had the highest deviations, ranging from 1.5% to 2.5%. Overall, the Standard models proved to be stable, and no pattern could be observed in relation to the size of the point clouds.

Regarding the Best architectures, it can be observed that in all cases, the average number of correct answers improved in relation to the Standard architecture. Furthermore, in almost all cases analyzed, the standard deviation was reduced, thus generating better and more stable models. The only case in which the standard deviation increased was in the Lamp-1 category with 2,048 points per cloud, obtaining a standard deviation of 0.93% in the Standard architecture and 1.13% in the Best architecture. Even so, the average accuracy in the Best architecture is higher and the difference between the standard deviations is small.

To facilitate understanding of the results, Figure 3 shows the boxplots of the loss and accuracy metrics in the test set for the Table-1 category, which has the largest amount of data available. First, it is noted that for the loss metric, the size of the point clouds does not directly affect the results obtained, both in the Standard architecture and in the Best architectures. It is also noted that the Best architectures present more consistent results, with less variation among the 10 experiments performed, in addition to achieving lower losses than the Standard architecture.

In terms of accuracy, the Best architectures achieved better results with less variation between models, but the difference is less significant than in the loss metric. In addition, the Best architecture with 1,024-point clouds did not achieve results as good as the other two point quantities, allowing for a discussion about the impact of point cloud size. A larger number of points provides a greater understanding of the object as a whole and the patterns between its parts, but at the same time, the object will have more details and the complexity will also be greater. Following this reasoning, the size of the point clouds tends not to greatly influence the results obtained.

Based on the discussions by (Qi et al., 2017), presented in Section 3, and the analyses carried out in this work, it can be seen that while the size of the point clouds is sufficient to understand the objects, this data tends not to greatly influence the performance of the models in point cloud processing tasks, such as classification and segmentation by parts. However, if the clouds are simplified to the extreme, the performance of the models will also be significantly impacted.



(a) Test Loss Boxplots (b) Test Accuracy Boxplots

Figure 3. Boxplots of Test Loss and Test Accuracy for Table-1 category

Finally, a statistical test is needed to prove the differences and equivalencies of the trained architectures. Considering only the averages of the executions in the accuracy metric, shown in Table 4, a 2% mean improvement was achieved in the Table-1, Chair-1, and Lamp-1 categories, and 6% in StorageFurniture-1. However, through statistical testing, it is possible to prove that these improvements are significant and that the models generated by the Best architectures are statistically better than those generated by the Standard one.

To do this, Critical Difference Diagrams were made with the Holm adjustment and an alpha of 0.05. The lower the values, the better the performance of the models. A line connecting two or more cases signifies their statistical equivalence. Thus, Figure 4 shows the critical difference diagram for the Table-1 category. For this category, the same behavior is observed for the two evaluation metrics. As a result, the Standard architecture performed equally well across all point cloud sizes. In addition, all Best architectures outperformed the Standard architecture, statistically confirming the analyses performed.

For the Table-1 category, it was also noted that the Best architecture for 512 points was statistically different from the others. In this case, despite the tendency for the size of the point clouds not to influence the results of the part segmentation task, as discussed earlier, some cases deviate from this rule. This situation may occur due to the complexity generated in this specific case, the Bayesian optimization performed, or the stochastic behavior of artificial neural networks, which we attempted to mitigate by executing 10 distinct runs in each study.

In the Chair-1 and StorageFurniture-1 categories, statistical differences were also observed between the Standard and Best architectures in the two evaluation metrics, again statistically confirming the analyses performed. Furthermore, as in the Table-1 category, statistical differences were also noted between the Best architectures and both Chair-1 and StorageFurniture-1, as well as between the Standard architecture. Thus, the same justifications apply, except for Bayesian optimization in the Standard architecture, which was not applied.

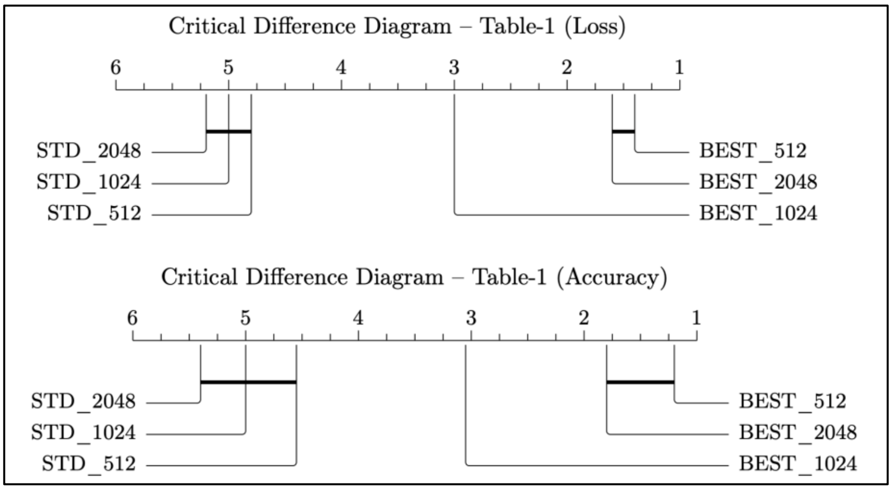


Figure 4. Critical Difference Diagrams for Table-1 category

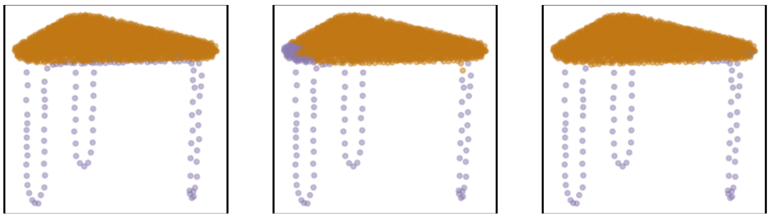
Finally, the Lamp-1 category behaved differently from the others, with different results in each of the evaluation metrics. In the loss metric, statistical differences were observed between the Standard and Best architectures, also statistically confirming the analyses performed. However, for the accuracy metric, all Best architectures were statistically equal to the standard architecture trained at 1,024 and 2,048 points. In addition, the Best architecture with 512 points was also statistically equivalent to Standard with 512 points. When analyzing Table 4, this behavior is explained by the close averages and the larger standard deviations between the architectures.

As a conclusion of the critical difference diagram analyses, statistical differences were found between the Standard and Best architectures in the Table-1, Chair-1, and StorageFurniture-1 categories. The only category in which statistical equivalences were observed was Lamp-1, which has the smallest number of instances available for model training and the largest number of parts that make up the objects. Therefore, the effectiveness of hyperparameter optimization in the part segmentation task depends on the complexity of the class, considering the number of parts of the objects and the number of instances available. It is possible that in cases with a lot of data for model training, the effectiveness of this optimization may also be lower. Even so, the optimization is performed to be at least equivalent to the baseline results and, therefore, is a strategy with considerable potential and many practical applications.

6.2 Qualitative Analysis

To evaluate the contribution of hyperparameter optimization qualitatively, the Table-1 and Lamp-1 classes were selected, both with 2,048 points, as they had the highest and lowest number of instances, respectively. As in both categories, the number of point clouds in the test set is too large for accurate manual analysis, therefore, the first 10 point clouds in each of the two categories were analyzed, comparing the ground truth with the Standard and Best architectures. It should be noted that to achieve a fairer comparison, the models with the lowest loss in the test were selected for this analysis.

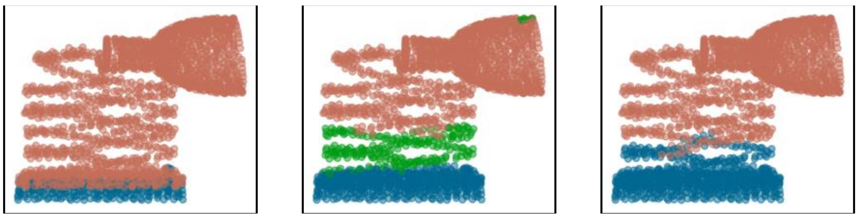
Figure 5 shows the best result found in the Table-1 category, with a considerable difference between both segmentations. It is noted that the Standard architecture had difficulty separating the top and legs of the table, incorrectly classifying part of the object. On the other hand, the Best architecture accurately understood the structure of the object, achieving a result visually equivalent to the ground truth. In addition, in the Table-1 category, there were two more instances with considerable improvements, two with smaller improvements and five other instances with visually equivalent results.



(a) 2,048 Points (b) Standard Architecture (c) Best Architecture

Figure 5. Comparison of results for a point cloud in the Table-1 category

Finally, Figure 6 shows the best result found in the Lamp-1 category, also showing a considerable difference between both segmentations. It can be observed that the Standard architecture incorrectly separated the object into three parts, while in the ground truth the object was divided into only two parts. This extra part can also be observed at the top right of the lamp, noting that the model in question was unable to fully understand the pattern of this object. On the other hand, the Best architecture divided the object into the two correct parts, better understanding the object's pattern. Still, the base of the lamp in this case should be lower, allowing for more future contributions. In addition, three instances obtained equivalent results in both architectures, one instance with slightly better results in the Best architecture, two instances slightly better in the Standard architecture, and three considerably better in the Standard architecture.



(a) 2,048 Points (b) Standard Architecture (c) Best Architecture

Figure 6. Comparison of results for a point cloud in the Lamp-1 category

Thus, it can be observed that although the average accuracy is higher in Best architectures, the impact of optimization will be unique in each instance. In addition, the amount of data for training the models, the number of parts in each category, and the overall complexity of the instances directly influence the results. For example, the Lamp-1 class has less data for training, a larger number of parts, and, in general, a more complex shape structure compared to the Table-1 category. Therefore, the average accuracy of the Table-1 category was higher.

7. CONCLUSIONS AND FUTURE WORK

Currently, the most widely used computational strategies for point cloud segmentation consist of artificial neural networks. However, these architectures are built generically to achieve satisfactory results across a wide range of object categories. Nevertheless, these generic architectures can be optimized for each specific case, thereby achieving better results. Therefore, this work sought to optimize the PointNet architecture, already well established in the literature, to build specific architectures for four categories of the PartNet dataset: Table-1, Chair-1, StorageFurniture-1, and Lamp-1. To this end, the architectures were optimized using the hyperparameter tuning technique through automated machine learning.

Furthermore, for each study category, this optimization process was performed on three point cloud sizes: 512, 1,024, and 2,048 points, to also analyze the impact of this item on both the original PointNet architecture and the optimized architectures. As a result of this work, we concluded that the size of the point clouds has a low impact on the results obtained in the point cloud part segmentation task, and it is not possible to observe a clear pattern in the behavior of this data. Regarding the architecture optimization process, satisfactory results were observed, achieving a 2% improvement in the test accuracy metric in the Table-1, Chair-1, and Lamp-1 categories, and a 6% improvement in the StorageFurniture-1 category.

It is interesting to note that, for each specific problem, a different architecture will be optimal, and it is recommended that this process be carried out for each case worked on. The choice of the number of points in each cloud will depend on the objective of each application, but the effectiveness of the hyperparameter tuning was shown to be similar in the cases tested. In addition, when analyzing the values of the hyperparameters selected in the best architectures, common behaviors were seen, such as the choice for the initial learning rate of 0.001.

For statistical verification of the improvements observed, critical difference diagrams were analyzed for each of the study categories. As a result, the Table-1, Chair-1, and StorageFurniture-1 categories were statistically better in their best architectures compared to the original PointNet architecture, presenting real contributions in this area of study. Qualitative analyses were also performed to analyze the results in the Table-1 and Lamp-1 categories. In the Table-1 category, different improvements were observed, mainly in a specific shape, which showed a significant improvement over the standard architecture. In the Lamp-1 category, positive and negative impacts were observed, highlighting that although the average results are superior in the optimized architecture, each instance will be impacted differently.

Finally, it is concluded that the hyperparameter optimization technique fulfilled the initial objectives, achieving more stable and accurate models. As a contribution, it was noticed that the results depend on the amount of data for training, the number of parts of each shape, as well as the complexity of each category structure. Therefore, the use of this process is recommended in future work within this area of research, enabling increasingly promising results to be achieved. As future work, we intend to carry out the study with larger numbers of points per cloud, different artificial neural network architectures, and evaluate other metrics, such as mIoU. Another improvement would be to use new categories available in the PartNet dataset, as well as other levels of detail, such as middle and fine-grained, which are already available too.

ACKNOWLEDGEMENT

Authors would like to thank the FAPESC agency and the CAPES foundation.

REFERENCES

Chang, A. X. et al., 2015. Shapenet: An information-rich 3D model repository. *ArXiv preprint arXiv:1512.03012*

Daif, H. and Marzouk, M., 2025. Point cloud classification and part segmentation of steel structure elements. *Neural Computing and Applications*, *37*(6), 4387-4407.

He, X., Zhao, K. and Chu, X., 2021. AutoML: A survey of the state-of-the-art. *Knowledge-based systems*, *212*, 106622.

Hutter, F., Kotthoff, L. and Vanschoren, J., 2019. Automated machine learning: Methods, systems, challenges. *Springer*.

Mo, K. et al., 2019. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 909-918.

Qi, C. R., Su, H., Mo, K. and Guibas, L., 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652-660.

Salehin, I. et al, 2024. AutoML: A systematic review on automated machine learning with neural architecture search. *Journal of Information and Intelligence*, *2*(1), 52-81.

Wang, Q. and Kim, M. K., 2019. Applications of 3D point cloud data in the construction industry: A fifteen-year review from 2004 to 2018. *Advanced engineering informatics*, *39*, 306-319.

Wang, X. et al, 2023. Recent advances in Bayesian optimization. *ACM Computing Surveys*, *55*(13s), 1-36.

Wielgosz, M., Puliti, S., Wilkes, P. and Astrup, R., 2023. Point2Tree (P2T) — Framework for parameter tuning of semantic and instance segmentation used with mobile laser scanning data in coniferous forest. *Remote Sensing*, *15*(15), 3737.

Wu, Z. et al., 2015. 3D shapenets: A deep representation for volumetric shapes. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1912-1920.

Zermas, D., Izzat, I. and Papanikolopoulos, N., 2017. Fast segmentation of 3D point clouds: A paradigm on LiDAR data for autonomous vehicle applications. *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 5067-5073.