

Benchmarking Deep Learning Models on Point Cloud Segmentation

Thanasis ZOUMPEKAS^{a,1}, Guillem MOLINA^a, Maria SALAMÓ^a and Anna PUIG^a

^a*Department of Mathematics and Computer Science, University of Barcelona, Barcelona, Spain*

Abstract. Point clouds are currently used for a variety of applications, such as detection tasks in medical and geological domains. Intelligent analysis of point clouds is considered a highly computationally demanding and challenging task, especially the segmentation task among the points. Although numerous deep learning models have recently been proposed to segment point cloud data, there is no clear instruction of which exactly neural network to utilize and then incorporate into a system dealing with point cloud segmentation analysis. Besides, the majority of the developed models emphasize more on the accuracy rather than the efficiency, in order to achieve great results. Consequently, the training, validation and testing phases of the models require a great number of processing hours and a huge amount of memory. These high computational requirements are commonly difficult to deal with for many users. In this article, we analyse five state-of-the-art deep learning models for part segmentation task and give meaningful insights into the utilization of each one. We advance guidelines based on different properties, considering both learning-related metrics, such as accuracy, and system-related metrics, such as run time and memory footprint. We further propose and analyse generalized performance metrics, which facilitate the model evaluation phase in segmentation tasks allowing users to select the most appropriate approach for their context in terms of accuracy and efficiency.

Keywords. Deep Learning, Neural Networks, Point Clouds, Segmentation, Analysis, Benchmark

1. Introduction

Nowadays, point cloud data is utilized as input for a great assortment of applications, including the creation of 3D models for manufactured parts, detection and identification procedures in medical, geological and autonomous-driving domains and engineering simulations [1]. A point cloud is a set of points in 3D space, characterized by the x , y , and z coordinates, that can be mainly acquired by 3D sensors, such as LIDARs and RGB-D cameras. These devices are able to capture with detail surface and geometrical properties of objects [2]. Indeed, point cloud data of a 3D object may contain millions of points with highly detailed information, but they also contain scattered, disjointed information and in most cases with a lot of noise [1].

¹Corresponding Author: Department of Mathematics and Computer Science, University of Barcelona, Barcelona, Spain; E-mail: thanasis.zoumpakas@ub.edu

In fact, the rapid advancements of the 3D sensing technology in recent times are increasing the demand for 3D point cloud processing techniques in order to extract implicit information. In this sense, deep learning approaches are the most frequently used techniques providing intelligent solutions in a wide range of applications. However, the intelligent analysis of such huge data is a highly computationally demanding and complex task, especially the segmentation task among the points [3]. The segmentation process is the categorization of the points of an object into different parts and groups. It is based on the notion that the points belonging to a specific area of an object have the same properties. The segmentation of scattered and huge data such as point clouds is a challenging process and usually segmentation models mainly focus on the improvement of accuracy [4] rather than efficiency.

Numerous intelligent approaches achieve great accuracy in segmenting parts of objects and whole scenes of point cloud data [5,6]. While some of them include measurements of the computational power per second (i.e. FLOPS), such as [6], or investigate memory costs of operations[7], there is still a big open question on the efficiency of the accurate models. It is considered highly important to include all feasible metrics for a proposed model, as pointed out by [8], but to the best of our knowledge, at the time of this writing, there is no specific performance evaluation approach that evaluates the point cloud data segmentation accuracy of a model concerning the time and memory allocation needed to achieve the best accuracy score. In this article, we address this issue and experiment with five of the most accurate Neural Networks² for segmenting parts of objects represented with point cloud data. Specifically, we analyse PointNet [9], PointNet++ [10], Kernel Point Convolution (KPConv) architecture [5], Position Pooling (PosPool) architecture [7] and Relation-Shape Convolutional (RSConv) neural network [6].

Thus, our paper makes three contributions: (i) We analyse the performance of the used models in relation to time and average memory allocation, apart from accuracy, (ii) We propose and formulate novel performance metrics taking into account both *learning-related*, such as *mIoU*, and *system-related* metrics, such as run time and memory footprint, (iii) We analyse in-depth these metrics to define clear and meaningful insights on the accuracy and efficiency of the models and the trade-off between them.

2. Related Work

Point cloud segmentation using deep learning models is an emerging study field and have recently caught the attention of a vast majority of researchers, mainly due to the presence of unique challenges, such as the lack of structure and the high dimensionality of the data [3]. Apart from that, numerous review studies present multiple deep learning models handling point clouds with a great accuracy [3,1,11]. The groundbreaking neural network dealing with such tasks was PointNet [9] and after its release, a great extent of models came out improving the segmentation and classification accuracy of it [11].

In the original study of PointNet, Qi et al. [9] include a short time and complexity analysis of it. However, their analysis do not take into consideration multiple comparison dimensions among the other published neural networks. More recently, Liu et al. [7], compare their proposed model to other studies showing a benchmark that is mostly fo-

²<https://paperswithcode.com/sota/3d-part-segmentation-on-shapenet-part>

cused on accuracy and endogenous parameters of the neural networks, for example, width and depth of the architectures. In addition, the majority of the new and state-of-the-art models, such as [12,5,10], focus primarily on the improvement of the segmentation accuracy, providing basic or no information regarding efficiency.

There are also approaches, that consider other metrics apart from accuracy of the model. Coleman et al. [13] present an evaluation approach of deep learning models taking into account *system-related* and *learning-related* performance metrics, such as training time and accuracy respectively. They provide a benchmark that compares the training time needed to achieve the state-of-the-art accuracy, as well as, the the time it takes to infer having the best accuracy. Additionally, an extensive research of time-to-accuracy performance analysis is presented in [14]. However, these studies focus on image classification and segmentation tasks, involving imagery data, i.e. 2D data but not point clouds.

On the other hand, Garcia-Garcia et al. [8] present an interesting study on deep learning techniques with an application to segmentation task. They put on foreground not only the accuracy but also the efficiency of a wide range of methods involving not just 2D image data but even 3D point cloud data. They highlight that there is a need for performance evaluation of segmentation models to be considered useful and valid in practice. Even though, they claim that the most important metrics for the evaluation of deep learning segmentation models should be the execution time, the memory footprint and the accuracy, stating that the trade-off among them should be parameterized depending on the system's or analysis' objective, their analysis provide only accuracy-related results.

We consider there is a gap in benchmarking in-depth segmentation models on point cloud data, taking into consideration dimensions such as execution time, memory footprint and accuracy, that allow users to validate and select the appropriate model for their specific application or context.

3. Proposal

We propose a performance benchmark for deep learning algorithms for the task of point cloud part segmentation. Our main inspiration is derived from [13,8] and the lack of studies highlighting which is the best Neural Network in terms of accuracy and efficiency for the part segmentation task. Please note, that by the term accuracy in the segmentation process, we denote the performance in the metric of point *Intersection over Union (IoU)*, which is explained onwards. Also, the term efficiency describes the least amount of resources of a method in run time and memory footprints. Therefore, we experiment with a well-known point cloud dataset, and we analyse five of the most accurate deep learning models providing not only segmentation accuracy, i.e learning-related metrics, but also system-related metrics, such as run time and memory footprints.

We acknowledge that accuracy is necessary in any intelligent system. However, other performance dimensions should also be considered and evaluated. Thus, we propose novel performance metrics aiming to balance the decision border between accuracy and efficiency, taking into consideration both learning-related and system-related metrics.

Considering learning-related metrics, the point *Intersection over Union (IoU)* performance metric is the most typical metric to evaluate 3D point cloud part segmentation models. *IoU* is simply defined as $IoU = \frac{A \cap B}{A \cup B}$, where A is the area of points of ground-

truth point cloud and B of the predicted point cloud. It appears in the majority of the research studies working in the aforementioned field, such as [9,10,5,7,6]. Specifically, two variants of the IoU metric are the most used ones, the mean Intersection over Union ($mIoU$) obtained by averaging across all Classes ($CmIoU$) and all Instances ($ImIoU$). For clarification, the $ImIoU$ is the average of the $mIoU$ across all point clouds, i.e. all learning instances. The $CmIoU$ metric is the average of the $mIoU$ across all point clouds belonging to specific classes and then averaged again on the number of classes. The details of the aforementioned metrics are also described in [6].

Aside from the learning-related performance metrics, we further measure the system-related metrics such as run time and average amount of memory allocated by the Graphics Processing Unit (GPU) of the whole learning procedure of each utilized neural network.

3.1. Proposed Metric

Specifically, we propose the F_{CmIoU} , F_{ImIoU} and $F_{general}$ and we analytically explain the formulation of them below, in equations (1), (2) and (3) respectively. We formulate the F_{CmIoU} (Eq. 1) that provides a per class segmentation accuracy ($CmIoU$) as well as the efficiency of the model involving the total run time (t_{total}), which is the total time spent to finish the whole learning process of training, validation and testing, and the average percentage of GPU memory allocation (GPU_{mem}) that is used for computations in the whole learning process. In a similar way, we formulate the F_{ImIoU} (Eq. 2), which aims to balance the decision border in model selection among $ImIoU$, t_{total} and GPU_{mem} . Finally, in an attempt to give a more generalized metric facilitating further the model selection procedure, considering both general accuracy and efficiency of the models, we propose the $F_{general}$ (Eq. 3), which is the arithmetic mean between the F_{CmIoU} and F_{ImIoU} . For comparison purposes across the different metrics, we normalize all the values per metric in the range $[0, 1]$.

In F_{CmIoU} (Eq. 1), we denote β the coefficient that balances the decision border among $CmIoU$, t_{total} and GPU_{mem} metrics. Higher β values portray more focus on the learning-related metric of $CmIoU$. In the notion of system-related metrics, i.e time and memory, the best model is considered to be the one having the lowest time duration and with the smallest memory footprint. Thus, in an attempt to highlight this, we formulate the time-related component in the equation as $(1 - t_{total})$ and the memory-related component as $(1 - GPU_{mem})$. Likewise, in F_{ImIoU} (Eq. 2) we denote α the balance coefficient among $ImIoU$, t_{total} and GPU_{mem} . Note that both β and α coefficients take values in the range of $[0, 1]$ and the sum of the weight factors in the equations is equal to 1, i.e. $\beta + \frac{(1-\beta)}{2} + \frac{(1-\beta)}{2} = 1$ and $\alpha + \frac{(1-\alpha)}{2} + \frac{(1-\alpha)}{2} = 1$. All the proposed metrics, i.e. F_{CmIoU} , F_{ImIoU} and $F_{general}$, take values in the range $[0, 1]$.

$$F_{CmIoU} = \beta * CmIoU + \frac{(1-\beta)}{2} * (1 - t_{total}) + \frac{(1-\beta)}{2} * (1 - GPU_{mem}) \quad (1)$$

$$F_{ImIoU} = \alpha * ImIoU + \frac{(1-\alpha)}{2} * (1 - t_{total}) + \frac{(1-\alpha)}{2} * (1 - GPU_{mem}) \quad (2)$$

$$F_{general} = \frac{F_{CmIoU} + F_{ImIoU}}{2} \quad (3)$$

Note that the proposed metrics can be used stand-alone (Eqs. 1 and 2) or combined in a unique metric (Eq. 3) for more generalized performance evaluation, depending on the objective of the application.

4. Evaluation

This section presents the segmentation models, the data used and the benchmark protocol in our evaluation process. Finally, we show and analyse our results.

4.1. Models

We investigate five of the most accurate neural networks dealing with 3D point cloud part segmentation. Following, we briefly describe them.

PointNet [9] is considered one of the fundamental and pioneering deep learning architectures on point cloud data, which handles both classification and segmentation tasks using point-wise operations. It focuses on the global structure of a point cloud while being symmetrical or invariant by the input order of point clouds. Additionally, it features point-wise robustness against noisy elements, perturbation or missing data. However, the point-wise feature extraction operations of PointNet do not take into consideration the topology of the points, i.e. neighbourhoods of points.

PointNet++ [10] is the successor of the PointNet architecture that is capable to focus on both the global and local structure of a point cloud object. Moreover, it also considers the geometrical properties and is robust to size variance and point density variance per point and per neighbourhood of points. One strong point of PointNet++ is that it is adequate to deal with surfaces of point clouds.

KPConv [5] approach emphasizes on local aggregation computations among the points. Specifically, it uses kernel operations, which are defined by a set of points distributed in a sphere. The convolutions are based on a defined metric system and are dependent on the dimensions of the input data. Thus, they need to be adjusted to adequate sizes for each problem. KPConv is one of the best state-of-the-art point cloud segmentation algorithms, which achieves great performance while being robust against varying density neighbourhoods.

PosPool [7] emphasizes also, like the KPConv architecture, on local aggregation computations among the points. It uses a more simple way of computing the local properties of point clouds rather than using complex structures. The authors claim that the PosPool neural network has similar computational capabilities and is more robust to noisy or missing data than most of its competitors, even by using zero learnable parameters for local aggregation operations.

RSCConv [6] architecture focus, on local aggregation computations. Equivalent to 2D convolutional operations are used to extract meaningful features from the input point cloud. They are defined by a spherical shaped neighborhood, which is characterized by subsets of input points as centroids. Mainly the point features are derived by a selected

relation, for instance the euclidean distance, between the points and the aforementioned centroids. RSCnv is considered to be a generalization of the traditional 2D convolutions, as the weights of the convolution matrix are dependent on the position of each point in relation to the center of the convolutional operation. Additionally, it is invariant to permutations, robust to rigid transformations and captures well the relations between the points.

It should be noted that all the aforementioned models evaluate their performance focusing mainly on learning-related metrics, such as *mIoU*.

4.2. Data

We use the ShapeNet dataset for part segmentation, namely ShapeNet-Part³. The utilized version of dataset contains 16881 models of 3D point clouds categorized in 16 distinct shapes. In each category, from two to six parts are annotated, summing up to 50 annotated parts in total. The labeled categories are: *aeroplane, bag, cap, car, chair, earphone, guitar, knife, lamp, laptop, motor, mug, pistol, rocket, skateboard and table*. More information on ShapeNet data can be found in its official and published articles [15,16].

4.3. Benchmark Protocol

We have created a benchmark protocol, according to which all the selected models have been trained, validated and tested. The utilized system for the experiments has the following configuration: (i) CPU: Intel Core i9-10900, (ii) RAM: 32GB, (iii) GPU: Quadro RTX 5000 - 16 GB, and (iv) OS: Ubuntu 20.04. In addition, we used the Torch-Points3D [17] framework, using Python 3.8.5, CUDA 10.2 and PyTorch 1.7.0 version.

Regarding the training, validating and testing of the utilized networks, we utilize the original split following the [15]. Specifically, we use a training set size of 12137 point clouds, a validation set size of 1870 and a test set size of 2874. The batch size is set to be 16 and the optimizer of the networks is the Adam. Additionally, to tackle overfitting issues, we use exponential learning rate decay and batch normalization techniques, both of them evaluated at every epoch. Finally, we trained all the networks for 200 epochs. After the training phase on each epoch, the models are validated and tested.

4.4. Results

Figure 1 displays the total run time of each model in relation to the total number of epochs and highlights when the best *CmIoU* and *ImIoU* are recorded in the testing phase of each one of the models. The first observation is that all the models require different run time per epoch to complete the learning process, with the fastest one being the PointNet++ and the slowest one the PPNet. Additionally, the models achieve their best accuracy in different epochs having distinct time duration in their learning processes.

Following our initial observations, we detail the measurements of the learning process of models in Table 1. We observe that RSCnv is the best method in test data achieving an *ImIoU* score of 85.47 and a *CmIoU* of 82.73. PointNet++ comes second in terms of *ImIoU* and *CmIoU* metrics, achieving scores of 84.93 and 82.50 respectively. Additionally, PPNet achieves the same *CmIoU* metric of 82.50 as PointNet++.

³Available at: https://shapenet.cs.stanford.edu/media/shapenet_part_seg_hdf5_data.zip

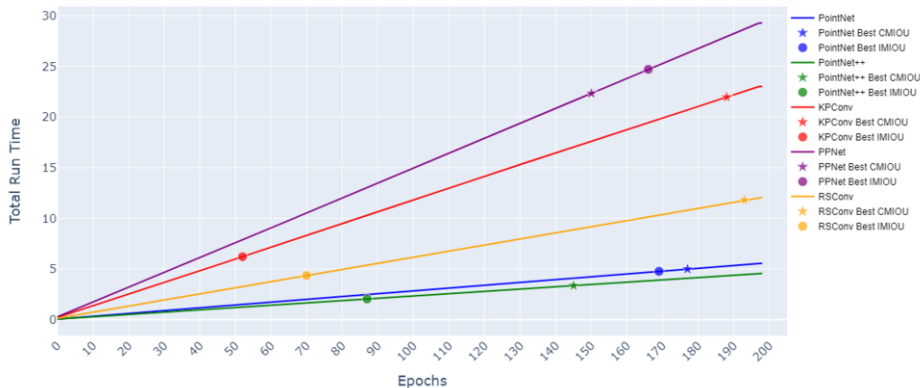


Figure 1. Comparison of Models - Time vs Epochs. The Total Run Time is denoted in hours.

Table 1. Deep Learning Models - Performance Evaluation of the learning process in 200 epochs. We display the best achieved $CmIoU$ and $ImIoU$ evaluation metrics in training, validation and test sets. We further denote the time spent to finish the whole learning process of training, validation and testing as $t_{total}(h)$, the time spent to achieve the best performance in relation to $CmIoU$ and $ImIoU$ metrics in the test data as $CmIoU_{time(h)}^{best}$, $ImIoU_{time(h)}^{best}$ respectively, and the average GPU memory allocation in percentage in whole learning process as $GPU_{mem}(\%)$. We use dark grey and light grey cell colors to denote the best and the second best score per metric respectively.

Method	Training Set		Validation Set		Test Set		$t_{total}(h)$	$ImIoU_{time(h)}^{best}$	$CmIoU_{time(h)}^{best}$	$GPU_{mem}(\%)$
	$ImIoU$	$CmIoU$	$ImIoU$	$CmIoU$	$ImIoU$	$CmIoU$				
PointNet	89.80	89.81	86.39	79.87	84.24	79.03	5.55	4.77	4.99	33.7
PointNet++	90.45	90.85	86.83	83.08	84.93	82.50	4.55	2.03	3.36	36.3
KPConv	91.99	91.91	86.61	82.43	84.22	82.39	23.01	6.21	21.96	60.4
PPNet	92.37	92.89	86.46	82.56	83.87	82.50	29.28	24.71	22.33	89.0
RSConv	91.53	92.39	87.12	82.82	85.47	82.73	12.06	4.35	11.81	55.5

An interesting observation is that PointNet++ is the fastest neural network with a total run time (t_{total}) of 4.55 hours, while being the second best in accuracy. It also achieves its peak performance in test data faster than all its competitors, i.e. $ImIoU_{time(h)}^{best} = 2.03$ and $CmIoU_{time(h)}^{best} = 3.36$ hours. RSConv and PointNet come second in $ImIoU_{time(h)}^{best}$ and $CmIoU_{time(h)}^{best}$ respectively.

In terms of average GPU memory usage, PointNet finishes in the first place with 33.7% although the second one, PointNet++, has approximately the same GPU memory usage with a value of 36.3%. Also, PPNet utilizes 89% of total GPU memory while needs about 29.28 hours of time for its computations, appearing to be the most inefficient in terms of memory and time spent. In addition, PPNet reaches the highest performance in learning-related metrics in training set with 92.37 in $ImIoU$ and 92.89 in $CmIoU$. However, its poor performance in test set may indicate overfitting issues. Considering our results, the first observation is as follows:

Observation 1. Learning-related and system-related metrics have different winners and the winners of one metric demonstrate lower scores on the others. Thus, a generalized metric that reflects the different behaviour of each model across all scores is needed.

Table 2. Deep Learning Models - Generalized Performance Evaluation. We use dark grey and light grey cell colors to denote the best and the second best score per metric respectively.

Method	$\beta = 0.3, \alpha = 0.3$			$\beta = 0.5, \alpha = 0.5$			$\beta = 0.8, \alpha = 0.8$		
	F_{ImIoU}	F_{CmIoU}	$F_{general}$	F_{ImIoU}	F_{CmIoU}	$F_{general}$	F_{ImIoU}	F_{CmIoU}	$F_{general}$
PointNet	0.76	0.69	0.72	0.61	0.49	0.55	0.38	0.20	0.29
PointNet++	0.88	0.96	0.92	0.82	0.96	0.89	0.73	0.95	0.84
KPConv	0.34	0.54	0.44	0.30	0.65	0.47	0.25	0.80	0.53
PPNet	0.00	0.28	0.14	0.00	0.47	0.23	0.00	0.75	0.38
RSConv	0.76	0.76	0.76	0.83	0.83	0.83	0.93	0.93	0.93

Our results suggest that certain methodology should be taken into consideration in order to select the best model, according to the use-case’s requirements, which is practically accurate but also efficient in time and memory needed for the learning procedure. As shown in Table 1, the most accurate model for part segmentation is the RSConv, according to the $ImIoU$ and $CmIoU$ metrics on the test dataset. Additionally, we observe that the most efficient model, taking into account the time and GPU memory allocation is PointNet++, while being second in terms of accuracy.

In Table 2 we evaluate our proposed metrics (see section 3.1). We choose to display three different scenarios simulating the final user’s needs. In the first scenario, we select values of $\alpha = 0.3$ and $\beta = 0.3$, which indicate that the model selection process will be more biased towards the system-related behaviour of the models. It is clear that PointNet++ architecture achieves the best scores of $F_{ImIoU} = 0.88$, $F_{CmIoU} = 0.96$, $F_{general} = 0.92$. In the second scenario, where $\alpha = 0.5$ and $\beta = 0.5$, we present a balanced approach, giving the same weight in learning- and system-related metrics of each model. In this case, RSConv comes first in F_{ImIoU} metric with a value of 0.83 although the PointNet++ performance is nearly equal with a $F_{ImIoU} = 0.82$. However, PointNet++ outperforms all the other methods in F_{CmIoU} and in $F_{general}$ metrics. Finally, the third scenario, where $\alpha = 0.8$ and $\beta = 0.8$, displays a learning-related model selection process. RSConv seems to be the prime, achieving the best F_{ImIoU} of 0.93 although comes second in F_{CmIoU} . However, according to the generalized metric $F_{general}$, RSConv maintains the highest score of 0.93. Our results raise a second observation:

Observation 2. *A generalized metric facilitates the model selection procedure according to the user’s needs, providing a meaningful insight on the trade-off between efficiency and accuracy.*

The three different scenarios presented, show the ability to select the best segmentation model according to an individual’s needs. For instance, adjusting the β and α values, to values higher than 0.5 indicates that accuracy is the main concern and values lower than 0.5 indicate that the process concerns efficiency more than accuracy. For a balanced performance evaluation procedure, aiming to select the model that balances accuracy and efficiency, a value of 0.5 should be selected.

To sum up, our initial results confirm, to an extent, the accuracy results, that appear in the literature, for each one of the models analyzed. One important aspect that is derived from our evaluation process is that the learning-related or system-related metrics if used in isolation do not provide clear instruction to a user on which is the best deep learning model. This clearly highlights the need for unified metrics, taking into account both learning- and system-related metrics.

Our findings set us apart from the majority of previous research in this field. While the work in [13,14] presents a time-to-accuracy analysis with a focus on image classification and segmentation tasks, i.e. 2D data, we provide an analysis involving accuracy, run time and memory footprint on point cloud data, i.e. 3D data. Other studies, such as [8], clearly indicate that is of utmost importance to evaluate deep learning segmentation models on the execution time, the memory footprint and the accuracy of them. However, their analysis presents only accuracy results of the utilized models. On the contrary, we analyse all of the aforementioned metrics, providing significant insights.

Actually, examining Figure 1, we could identify some additional remarks, such as (i) there is a difference between the epochs where the best $CmIoU$ and $ImIoU$ are obtained, and (ii) the PPNet model is the only model of the five that we have investigated, that achieved first the best $CmIoU$ and second the $ImIoU$. Potential next steps are to enrich our proposed metrics by focusing on exploiting this information and taking into account these additional factors and to further analyse the features of each class of the ShapeNet dataset to include the concept of per-point classification accuracy. We expect that the inclusion of all this additional information in our metrics and the rigorous experimentation with more available datasets for the point-cloud part segmentation task may produce new insights into the models.

5. Conclusion

The majority of the research studies in part segmentation analysis using point clouds mostly focus on the improvement of the learning-related metrics of $CmIoU$ and $ImIoU$, i.e. accuracy, and there is a little emphasis on the system-related metrics, i.e. efficiency, and in the trade-off between them. Therefore, the scope of this article has been limited to the analysis of learning-related and system-related performance metrics, which extends considerably the point of view of the current literature in point cloud part segmentation.

This paper provides experimental insights aiming to balance the decision border of the model selection procedure in point cloud part segmentation methodologies. We have analysed five neural network models, namely PointNet, PointNet++, KPConv, PPNet, RSConv, using the mostly utilized ShapeNet dataset in the task of 3D point cloud part segmentation. The first outcome of our analysis is that the most accurate model seems to be the RSConv architecture and the most efficient the PointNet++. We propose the F_{CmIoU} , F_{ImIoU} , and the arithmetic mean of those $F_{general}$. Our metrics aid the benchmarking of different models and provide insights on the trade-off between accuracy and efficiency. We conclude that in a case with a balanced trade-off between accuracy and efficiency, i.e. $\alpha = 0.5$ and $\beta = 0.5$, the selected model should be PointNet++, which is the most efficient while achieves the second highest $CmIoU$ and $ImIoU$ values in test data. Note that our proposed metrics portray the ability to personalize the model selection, by selecting the trade-off between accuracy and efficiency depending on the research objective.

We believe that there are possible further exploration directions, such as the development of a benchmark taking into consideration varying comparison dimensions. As future work, we consider that additional factors should be included in the metrics, such as robustness.

Acknowledgements



This project has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 860843.

References

- [1] Bello SA, Yu S, Wang C, Adam JM, Li J. Review: Deep learning on 3D point clouds. MDPI AG; 2020. doi:10.3390/rs12111729.
- [2] Wang Q, Tan Y, Mei Z. Computational Methods of Acquisition and Processing of 3D Point Cloud Data for Construction Applications. *Archives of Computational Methods in Engineering*. 2020 4;27(2):479–499. doi:10.1007/s11831-019-09320-4.
- [3] Guo Y, Wang H, Hu Q, Liu H, Liu L, Bennamoun M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Transactions on PAMI*. 2020;1. doi:10.1109/tpami.2020.3005434.
- [4] Nguyen A, Le B. 3D point cloud segmentation: A survey. In: 2013 6th IEEE Conf. on Robotics, Automation and Mechatronics (RAM). IEEE; 2013. p. 225–230. doi:10.1109/RAM.2013.6758588.
- [5] Thomas H, Qi CR, Deschaud JE, Marcotegui B, Goulette F, Guibas L. KPConv: Flexible and deformable convolution for point clouds. In: *Proceedings of the IEEE Int. Conf. on Computer Vision*. vol. 2019-Octob. IEEE Inc.; 2019. p. 6410–6419. doi:10.1109/ICCV.2019.00651.
- [6] Liu Y, Fan B, Xiang S, Pan C. Relation-shape convolutional neural network for point cloud analysis. In: *Proceedings of the IEEE Computer Society Conf. on CVPR*. vol. 2019-June. IEEE Computer Society; 2019. p. 8887–8896. doi:10.1109/CVPR.2019.00910.
- [7] Liu Z, Hu H, Cao Y, Zhang Z, Tong X. A Closer Look at Local Aggregation Operators in Point Cloud Analysis. In: *Lecture Notes in Computer Science*. vol. 12368 LNCS; 2020. p. 326–342. doi:10.1007/978-3-030-58592-1_20.
- [8] Garcia-Garcia A, Orts-Escolano S, Oprea S, Villena-Martinez V, Martinez-Gonzalez P, Garcia-Rodriguez J. A survey on deep learning techniques for image and video semantic segmentation. Elsevier Ltd; 2018. doi:10.1016/j.jasoc.2018.05.018.
- [9] Qi CR, Su H, Mo K, Guibas LJ. PointNet: Deep learning on point sets for 3D classification and segmentation. In: *Proceedings - 30th IEEE Conf. on CVPR 2017*. vol. 2017-Janua. IEEE Inc.; 2017. p. 77–85. doi:10.1109/CVPR.2017.16.
- [10] Qi CR, Yi L, Su H, Guibas LJ. PointNet++: Deep hierarchical feature learning on point sets in a metric space. In: *Advances in Neural Information Processing Systems*. vol. 2017-Decem. Neural information processing systems foundation; 2017. p. 5100–5109. doi:10.5555/3295222.
- [11] Liu W, Sun J, Li W, Hu T, Wang P. Deep learning on point clouds and its application: A survey. MDPI AG; 2019. doi:10.3390/s19194188.
- [12] Peyghambarzadeh SMM, Azizmalayeri F, Khotanlou H, Salarpour A. Point-PlaneNet: Plane kernel based convolutional neural network for point clouds analysis. *Digital Signal Processing*. 2020 3;98:102633. doi:10.1016/j.dsp.2019.102633.
- [13] Coleman C, Narayanan D, Kang D, Zhao T, Zhang J, Nardi L, et al. DAWNbench: An End-to-End Deep Learning Benchmark and Competition. 31st Conf on Neural Information Processing Systems (NIPS 2017). 2017. Available from: <http://dawn.cs.stanford.edu/benchmark>.
- [14] Coleman C, Kang D, Narayanan D, Nardi L, Zhao T, Zhang J, et al. Analysis of dawnbench, a time-to-accuracy machine learning performance benchmark. *Operating Systems Review (ACM)*. 2019 7;53(1):14–25. doi:10.1145/3352020.3352024.
- [15] Chang AX, Funkhouser T, Guibas L, Hanrahan P, Huang Q, Li Z, et al. ShapeNet: An Information-Rich 3D Model Repository. arXiv. 2015 12. Available from: <http://arxiv.org/abs/1512.03012>.
- [16] Yi L, Kim VG, Ceylan D, Shen IC, Yan M, Su H, et al. A scalable active framework for region annotation in 3D shape collections. *ACM Transactions on Graphics*. 2016;35(6). doi:10.1145/2980179.2980238.
- [17] Chaton T, Chaulet N, Horache S, Landrieu L. Torch-Points3D: A Modular Multi-Task Framework for Reproducible Deep Learning on 3D Point Clouds. In: *Proceedings - 2020 Int. Conf. on 3D Vision, 3DV 2020*. IEEE Inc.; 2020. p. 190–199. doi:10.1109/3DV50981.2020.00029.