

Survey and Evaluation of Neural 3D Shape Classification Approaches

This paper was downloaded from TechRxiv (<https://www.techrxiv.org>).

LICENSE

CC BY 4.0

SUBMISSION DATE / POSTED DATE

26-04-2021 / 06-05-2021

CITATION

Mirbauer, Martin; Krabec, Miroslav; Křivánek, Jaroslav; Šikudová, Elena (2021): Survey and Evaluation of Neural 3D Shape Classification Approaches. TechRxiv. Preprint.
<https://doi.org/10.36227/techrxiv.14447250.v1>

DOI

[10.36227/techrxiv.14447250.v1](https://doi.org/10.36227/techrxiv.14447250.v1)

Survey and Evaluation of Neural 3D Shape Classification Approaches

Supplementary Material

Martin Mirbauer, Miroslav Krabec, Jaroslav Křivánek, Elena Šikudová

1 OUR SHAPENETCORE SPLIT

Due to uneven distribution of training and testing subsets within categories, duplicates and some models being present in multiple categories in the official ShapeNetCore split, we generated our split where each category has 80% of objects used for training and 20% for testing. Table 1 shows the category sizes and train/test subset sizes. The final split can be downloaded from the project webpage.¹

TABLE 1
List of ShapeNetCore categories and the number of training and test models in each category in our split

Category	Train / Test	Category	Train / Test
table	6700 / 1676	pistol	212 / 53
chair	5276 / 1319	telephone	206 / 52
airplane	3235 / 809	piano	191 / 48
car	2811 / 703	bed	175 / 44
sofa	2403 / 601	stove	174 / 44
rifle	1864 / 467	mug	171 / 43
lamp	1853 / 464	bowl	138 / 35
vessel	1550 / 388	printer	132 / 33
bench	1386 / 347	washer	132 / 34
loudspeaker	1273 / 319	helmet	129 / 33
cabinet	1237 / 310	skateboard	121 / 31
display	863 / 216	microwave	120 / 31
bus	751 / 188	tower	98 / 25
bathhtub	683 / 171	camera	90 / 23
cellular	663 / 166	can	84 / 21
telephone	637 / 160	basket	81 / 21
guitar	593 / 149	pillow	76 / 20
faucet	514 / 129	mailbox	74 / 19
clock	439 / 110	dishwasher	72 / 19
pot	438 / 110	rocket	68 / 17
jar	381 / 96	bag	65 / 17
bottle	360 / 91	birdhouse	58 / 15
laptop	352 / 89	earphone	58 / 15
bookshelf	339 / 85	microphone	53 / 14
knife	311 / 78	remote	52 / 14
motorcycle	269 / 68	control	51 / 13
ashcan	257 / 65	computer	44 / 12
file	228 / 58	keyboard	44 / 12
(continues in the next column)		cap	44 / 12
		Total	40591 / 10178

2 COMPARISON OF DIFFERENT DATA CONVERSION METHODS

Here we show outputs of different conversion methods for each representation. The model *airplane_0627* from ModelNet40 is used as an example 3D shape. Some original conversion tools were not included in our framework because they use of commercial software ([27] and [44]) or they were not publicly available ([60]).

Figure 1 shows slight differences in the voxelization result of the OpenVDB library, which we used in our conversion, compared to the original voxelization provided by Brock et al. [27].



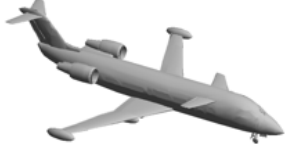
(a) Original voxel representation provided by authors of VRN (b) Our voxelization using OpenVDB

Fig. 1. Illustration of voxel representation

For image-based multi-view approaches, we render three kinds of images: one using a physically based renderer *PBRT* [167] using a perspective (not orthographic) camera projection, and *shaded* and *depth* images using code by Su et al. [13]. Outputs of these rendering methods are shown in Figure 2, compared to the pre-rendered subset of ModelNet40 provided by Su et al. [44]. Across all three datasets used in our experiments, the *shaded* method yields the best results: on average 0.59 pp better than *depth*, which is on average better by 0.68 pp than *pbrt*.

The methods we used for point cloud generation by sampling a mesh surface also affect the achieved performance: the *uniform* sampling reached the highest maximum accuracy, followed by *lloyd* and *sobol*, but did not reach the

1. <https://cg.mff.cuni.cz/~martinm/papers/2021-survey-eval>



(a) Phong-shaded image from [44]



(b) Our *PBRT* rendering



(c) *depth* image rendered using code from [13]



(d) *shaded* image rendered using code from [13]

Fig. 2. Illustration of differently rendered airplane_0627 model

performance of the authors-provided point clouds created using farthest point sampling. Figure 3 shows a visualization of the different point cloud sampling methods.

3 TRAINING PARAMETERS

The most important parameters we used to train the neural networks were set as follows:

- *vrn*
training epochs: 20
batch size: 24
learning rate: 0.002 for 10 epochs and then 0.0002
number of rotations: 24
- *octree*
training epochs: 50
batch size: 64
learning rate: 0.1, divided by ten every ten epochs
number of rotations: 12
- *octree-adaptive*
training epochs: 50
batch size: 64
learning rate: 0.1, divided by ten every ten epochs
number of rotations: 12
- *vgg*
training epochs: 20
batch size: 60
learning rate: 0.0001, multiplied by 0.75 every three epochs
number of views: 12
- *mvccnn2*
training epochs: 30+30
batch size: 64



(a) Original point cloud provided by the authors of PointNet

(b) Uniform sampling



(c) Lloyd sampling



(d) Sobol sequence sampling

Fig. 3. Illustration of point cloud representations, each sampling contains 2048 points

learning rate: 0.00005
number of views: 12

- *rotnet*
training epochs: 200
batch size: 40
learning rate: 0.0001 divided by ten every fifty epochs
number of views: 12
- *seq2seq*
training epochs: 200
batch size: 32
learning rate: 0.0002
number of views: 12
- *pointnet*
training epochs: 200
batch size: 64
number of points: 2048
learning rate: 0.0001 multiplied by 0.8 every 20 epochs
number of rotations: 12
- *pointnet2*
training epochs: 200
batch size: 32
number of points: 2048
learning rate: 0.0001 multiplied by 0.7 every 20 epochs
number of rotations: 12
- *sonet*
training epochs: 400
batch size: 8
number of points: 5000
learning rate: 0.001 divided by two every 40 epochs
number of rotations: 1
- *kdnet*

TABLE 2

Table of approximate training times on ModelNet40 (time spent training one epoch) and approximate sizes of the saved models, roughly corresponding to the number of trainable parameters of the model

Network	Epoch time [min]	Model size [MB]
Volumetric grid		
<i>vrn</i>	407	52
<i>octree</i>	5.5	2.5
<i>octree-adaptive</i>	1.5	2.5
Multi-view		
<i>mvnncnn2</i>	12.5	510
<i>rotnet</i>	6	230
<i>vgg</i>	29	550
<i>seq2seq</i>	0.6	30
Point cloud		
<i>pointnet</i>	4.5	40
<i>pointnet2</i>	2	17
<i>sonet</i>	1.5	10
<i>kdnet</i>	3	8

training epochs: 200
 batch size: 16
 number of points: 2048
 learning rate: 0.001
 number of rotations: 12

4 RESULTS

The results of our experiments are shown in Figure 6. In Table 3 we present details about the distribution of the achieved accuracies during our experiments.

4.1 Relationship of achieved accuracy and computational cost

In Table 2 we show run times and sizes of the stored model for each network. Figures 4 and 5 show the relationships of accuracy and time or model size graphically.

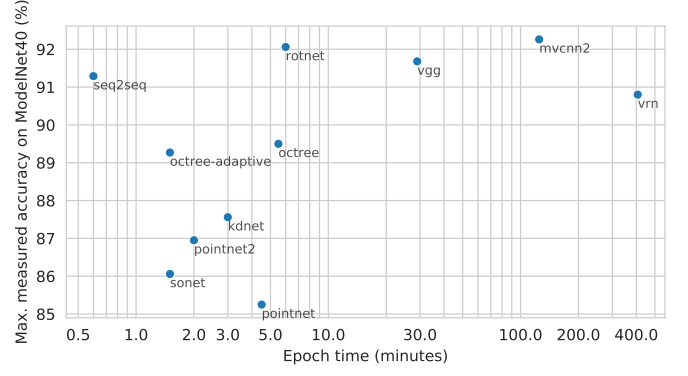


Fig. 4. Relationship of the achieved accuracy and training time.

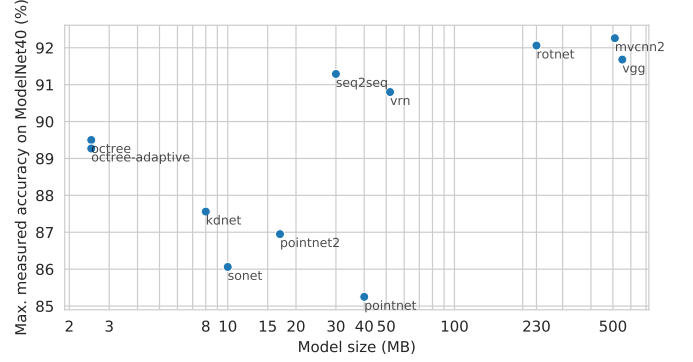


Fig. 5. Relationship of the achieved accuracy and model size.

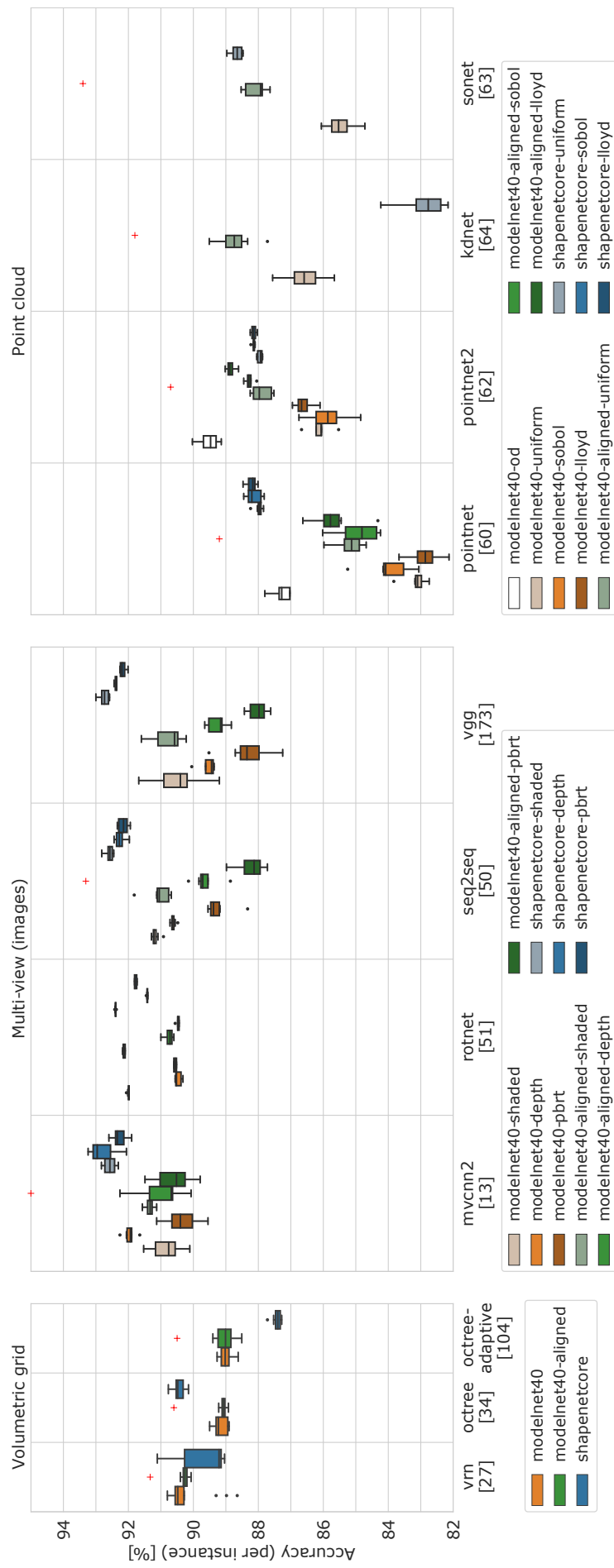


Fig. 6. The measured accuracies. For each network x dataset (x dataset variant), a distribution (boxplot) of accuracies on the test set in the best epoch and 3 before/after. • marks outliers and + marks the reported accuracy on ModelNet40 (official train/test split).

TABLE 3: Minimum, maximum and mean accuracies (%) and standard deviation (in percentage points) of the measured accuracies around the epoch with the highest test accuracy. The suffix *od* marks “original data” (provided by respective authors)

representation	net	dataset	accuracy			
			min	max	mean	std
Multi-view (images)	mvcnn2	modelnet40-aligned-depth	90.07	92.26	91.00	0.72
		modelnet40-aligned-pbrt	89.79	91.49	90.62	0.61
		modelnet40-aligned-shaded	91.13	91.57	91.34	0.14
		modelnet40-depth	91.65	92.26	91.96	0.20
		modelnet40-pbrt	89.55	91.13	90.35	0.53
		modelnet40-shaded	90.11	91.53	90.84	0.50
		shapenetcore-depth	92.06	93.24	92.79	0.44
		shapenetcore-pbrt	91.90	92.60	92.27	0.25
		shapenetcore-shaded	92.31	92.83	92.57	0.21
	rotnet	modelnet40-aligned-depth	90.60	91.00	90.75	0.14
		modelnet40-aligned-pbrt	90.44	90.56	90.47	0.04
		modelnet40-aligned-shaded	92.10	92.18	92.13	0.04
		modelnet40-depth	90.32	90.56	90.46	0.11
		modelnet40-pbrt	90.52	90.60	90.56	0.04
		modelnet40-shaded	91.98	92.06	92.00	0.03
		shapenetcore-depth	91.40	91.45	91.42	0.02
		shapenetcore-pbrt	91.73	91.81	91.78	0.04
		shapenetcore-shaded	92.37	92.42	92.39	0.02
	seq2seq	modelnet40-aligned-depth	88.86	90.15	89.62	0.39
		modelnet40-aligned-pbrt	87.72	88.98	88.23	0.44
		modelnet40-aligned-shaded	90.68	91.82	91.04	0.39
		modelnet40-depth	90.48	90.72	90.62	0.08
		modelnet40-pbrt	88.33	89.55	89.23	0.42
		modelnet40-shaded	90.92	91.29	91.17	0.13
		shapenetcore-depth	91.97	92.44	92.26	0.16
		shapenetcore-pbrt	91.94	92.34	92.16	0.16
		shapenetcore-shaded	92.45	92.82	92.58	0.13
	vgg	modelnet40-aligned-depth	88.83	89.65	89.28	0.30
		modelnet40-aligned-pbrt	87.62	88.43	88.02	0.31
		modelnet40-aligned-shaded	90.22	91.60	90.79	0.49
		modelnet40-depth	89.36	90.05	89.57	0.32
		modelnet40-pbrt	87.25	89.52	88.31	0.70
		modelnet40-shaded	89.20	91.68	90.39	0.70
		shapenetcore-depth	92.36	92.44	92.39	0.04
		shapenetcore-pbrt	92.01	92.26	92.17	0.10
		shapenetcore-shaded	92.58	93.00	92.74	0.16
Point cloud	kdnet	modelnet40-aligned-uniform	87.72	89.51	88.71	0.57
		modelnet40-uniform	85.66	87.56	86.58	0.61
		shapenetcore-uniform	82.16	84.23	82.89	0.72
	pointnet	modelnet40-aligned-lloyd	84.32	86.63	85.67	0.71
		modelnet40-aligned-sobol	84.24	86.02	84.92	0.68
		modelnet40-aligned-uniform	84.68	85.98	85.18	0.44
		modelnet40-lloyd	82.13	83.67	82.88	0.49
		modelnet40-od	87.03	87.80	87.27	0.28
		modelnet40-sobol	83.06	85.25	83.96	0.71
		modelnet40-uniform	82.74	83.83	83.14	0.34
		shapenetcore-lloyd	88.01	88.47	88.21	0.16
		shapenetcore-sobol	87.82	88.45	88.13	0.25
		shapenetcore-uniform	87.84	88.24	87.98	0.13
	pointnet2	modelnet40-aligned-lloyd	88.61	89.02	88.85	0.17
		modelnet40-aligned-sobol	88.05	88.45	88.27	0.16
		modelnet40-aligned-uniform	87.52	88.25	87.89	0.31

Volumetric grid	sonet	modelnet40-lloyd	86.10	86.95	86.60	0.36
		modelnet40-od	89.14	90.03	89.52	0.33
		modelnet40-sobol	84.85	86.75	85.87	0.63
		modelnet40-uniform	85.53	86.67	86.11	0.41
		shapenetcore-lloyd	88.03	88.25	88.14	0.08
		shapenetcore-sobol	88.10	88.23	88.15	0.06
		shapenetcore-uniform	87.87	88.05	87.95	0.08
		modelnet40-aligned-uniform	87.64	88.53	88.09	0.34
		modelnet40-uniform	84.72	86.06	85.48	0.44
		shapenetcore-uniform	88.47	88.97	88.67	0.19
	octree	modelnet40	88.90	89.50	89.16	0.24
		modelnet40-aligned	88.92	89.22	89.07	0.10
		shapenetcore	90.15	90.77	90.44	0.20
	octree-adaptive	modelnet40	88.62	89.27	89.00	0.22
		modelnet40-aligned	88.51	89.40	89.01	0.31
		shapenetcore	87.28	87.72	87.43	0.15
	vrn	modelnet40	88.65	90.80	90.19	0.68
		modelnet40-aligned	90.07	90.40	90.25	0.14
		shapenetcore	89.04	91.11	89.74	0.94
