A Large-scale Annotated Mechanical Components Benchmark for Classification and Retrieval Tasks with Deep Neural Networks

Sangpil Kim[*1], Hyung-gun Chi[*1], Xiao Hu[1], Qixing Huang[2], and Karthik Ramani[1]

[1] Purdue University, West Lafayette, IN 47907, USA
{kim2030, chi45, hu440, ramani}@purdue.edu
[2] Austin University, Austin, TX 78712, USA
huangqx@cs.utexas.edu

Abstract. We introduce a large-scale annotated mechanical components benchmark for classification and retrieval tasks named Mechanical Components Benchmark (MCB): a large-scale dataset of 3D objects of mechanical components. The dataset enables data-driven feature learning for mechanical components. Exploring the shape descriptor for mechanical components is essential to computer vision and manufacturing applications. However, not much attention has been given on creating annotated mechanical components datasets on a large-scale. This is because acquiring 3D models is challenging and annotating mechanical components requires engineering knowledge. Our main contributions are the creation of a large-scale annotated mechanical component benchmark, defining hierarchy taxonomy of mechanical components, and benchmarking the effectiveness of deep learning shape classifiers on the mechanical components. We created an annotated dataset and benchmarked seven state-of-the-art deep learning classification methods in three categories, namely: (1) point clouds, (2) volumetric representation in voxel grids, and (3) view-based representation.

Keywords: Deep Learning, Mechanical Components, Benchmark, 3D Objects, Classification, Retrieval

1 Introduction

The application of machine learning is highlighted recently due to the improved effectiveness of the deep neural networks [12, 16, 21, 32, 36, 37]. Along with deep neural networks, data driven algorithms and the creation of a large-scale datasets [4,8,26,47] have led to a series of breakthroughs in computer vision [6,14,45] and graphics [9,23,33]. The development of ImageNet [8], which used the class hierarchical structure from WordNet [31] to maximize the dataset coverage, showed
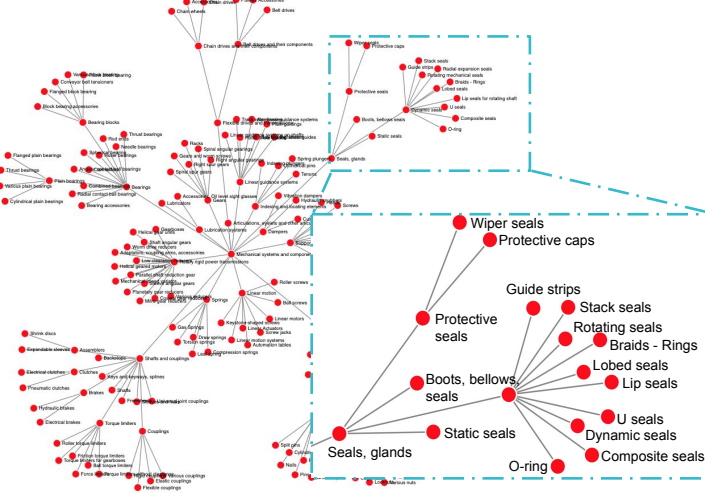
---

[*]These authors made an equal contribution.

Fig. 1. The hierarchy taxonomy of mechanical components based on the International Classification for Standards.

that a well-structured and annotated dataset is crucial for developing geometric feature descriptors. The pre-trained deep neural network descriptors using ImageNet have been widely used to extract low-dimensional representations that are used in tasks of object detection [25, 28, 37], semantic segmentation [5, 15, 34, 39], image caption generator [19, 46], and image retrieval [20, 43].

The creation of a large-scale mechanical components dataset with well-organized hierarchical classes and annotations is needed for developing and benchmarking geometric feature descriptors in the manufacturing industry [10, 18]. Geometric features extracted from the descriptors are fundamental cues to retrieve objects given the query object and classifying objects given image, volumetric representation, or point clouds.

However, in the manufacturing, design, and supply chain areas, the classification of mechanical components with deep neural networks has not been addressed due to the lack of large-scale annotated datasets. Without a standardized dataset, it is difficult to develop and compare learning algorithms on mechanical components [42].

Creating a large-scale mechanical component dataset is challenging due to the significant difficulty of collecting 3D CAD models of mechanical components. Different from common-object datasets [4, 32, 42, 47], the accessibility of most mechanical components is limited because of proprietary and ownership issues with specially designed models. Products and manufacturing models are held by companies for commercial usages, resulting in a deficiency of open-source components datasets. The inconsistency and incompatibility of mechanical components from available sources require massive effort on filtering and annotating
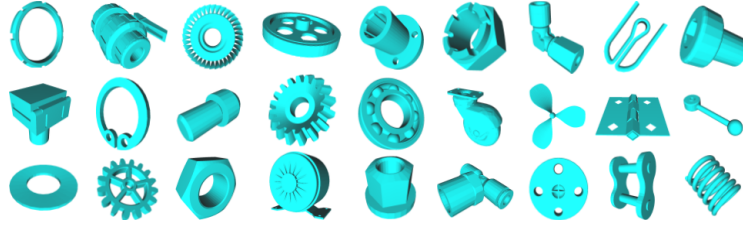
Fig. 2. Randomly sampled mechanical components from the MCB.

the data. Also, annotating mechanical components is harder than common objects since it demands more knowledge and expertise from annotators to properly annotate engineering components.

To resolve this difficulty, we established a hierarchical semantic taxonomy as a guideline based on the International Classification for Standards (ICS) published by the International Organization for Standardization (ISO). The tree structure of our proposed hierarchical taxonomy is depicted in Figure 1. Details are provided in the supplementary document. To collect annotations, we developed a web application to reduce the difficulty of filtering and annotating (Figure 4). This application supports controllable viewing, displaying meta-information, annotation, and filtering parts by viewing multiple parts as a tabular to visually see the consistent shape features with in the same class rather than viewing each individual part. These functionalities make the benchmark creation faster and more accurate for fine-grained categories.

Furthermore, we benchmark seven state-of-the-art shape descriptors to analyze the properties of mechanical components. Seven methods are carefully selected from three different 3D object representations: (1) point clouds, (2) voxel girds, and (3) view-based. From the benchmark results, the input representation is not the core factor that determines the performance. DLAN [11], which uses voxel grids, and PointCNN [24], which uses a point cloud input representation that focuses on local shape features, perform relatively well on both retrieval and classification tasks. The view-based methods are not robust on unseen orientation in shape retrieval tasks, which is also observed in common object retrieval tasks [41]. However, the descriptors [11,36,44] show significantly different results from common object classification tasks, which indirectly indicates that topological and geometrical characteristics of mechanical components are different from common objects. We report micro- and macro-precision, recall, F-score, mAP, and NDCG to evaluate retrieval tasks. For classification tasks, we report accuracy per class, accuracy per instance, F1 score, and average precision.

Our main contributions are as follows:

1. We create a hierarchical taxonomy based on the International Classification for Standards.
2. We annotate and collect a large-scale mechanical components benchmark. The mechanical components are annotated by mechanical engineering experts.

3. We benchmark seven deep learning-based classifiers and analyze their performances with the mechanical components dataset.
4. We develop a web-based framework which has a viewing, annotating, and filtering featurtovere to guide annotators.

Each contribution above provides a significant benefit to the computer vision community and opportunities for researchers to develop new algorithms for the mechanical components.

2   Related works

We will explain existing common-object and mechanical components datasets for 3D geometric data-driven learning methods. We summarized the overview of these datasets in Table 1. The reviews of shape classification and retrieval are detailed in Section 5.

Table 1. Comparison table of the MCB dataset with other datasets. CO and MC stands for common objects and mechanical components, respectively. ShapeNetCore, ShapeNetSem, and PartNet use models from the ShapeNet.

Dataset	# Class	# Models	Type
ModelNet [47]	40	12,311	CO
ShapeNet [4]	3,135	+3,000,000	CO
ShapeNetCore	55	51,300	CO
ShapeNetSem	270	12,000	CO
PrincetonSB [42]	92	6,670	CO
PartNet [32]	24	26,671	CO
ABC [11]	N/A	+1,000,000	MC
AAD [2]	9	180	MC
ESB [18]	45	867	MC
MCB (Ours)	68	58,696	MC

Large-scale 3D object datasets The Princeton Shape Benchmark (PSB) [42] is an early work that collected and annotated 3D objects for shape matching and classification benchmarking. It collected 3D polygonal models from the World Wide Web and classified them based on the method of construction, such as man-made and natural objects. ShapeNet [4] is a large-scale dataset of high-quality 3D models of objects, which are widely used in various tasks such as instance segmentation [32], shape retrieval [41], and shape reconstruction [7]. ModelNet [47] consists of two datasets (a 10-class dataset and a 40-class dataset) and demonstrates a comprehensive clean collection of 3D CAD models of objects. PartNet [32] is a fine-grained, instance-level, hierarchical parts dataset. It used 3D objects from ShapeNet and was annotated by 66 annotators.

Engineering shape datasets Engineering shape datasets have been developed to improve the shape-based retrieval of 3D data [17]. The Engineering Shape Benchmark (ESB) [18] is an annotated engineering shape dataset. It proposed an approach that defines the class by mechanical part's name—not by functionality—and benchmarked analytical shape descriptor. However, the number of models in the ESB dataset is not sufficient for training a robust feature extractor, and classes are only classified by their shape, which limits the usage of the dataset. The Actual Artifacts Dataset (AAD) [2] consists of four datasets with a total around 700 models and provides several classifications for engineering artifacts selected from the National Design Repository (NDR) [38]. Recently, A Big Cad (ABC) Model Dataset [22] proposed one million Computer-Aided Design (CAD) models dataset without annotations.

3   Properties of mechanical components

Mechanical components, shown in Figure 2, have sharp edges, well-defined surfaces, and high genus, which distinguish them from common objects. Since the shape of mechanical parts represents their physical functions, the functionality of machine elements is sensitive to small details, resulting difficulty in annotation. Therefore, mechanical components are often categorized by their detailed shape, whereas common objects are mainly identified by their general shape. The shape and location of detail features often determine the function of engineering parts.

Fig. 3. Examples of detail features making categorical changes.

The shape of the detail features and the function of engineering parts are usually interdependent. For example, the only difference in shape between a thrust washer and a lock washer is the split detail feature, as seen in Figure 3 (a), but they possess distinct functionality. A thrust washer spreads fastener loads, while a split lock washer uses the split feature to lock a nut and bolt in place. In another case, a hex nut and a lock nut share a hexagonal shape. However, the lock nut has an additional circular feature that houses a nylon insert, as seen in Figure 3 (b). A hex nut mates with a bolt to fasten materials together, and while the lock nut performs a similar function, the nylon insert keeps the nut from coming loose from the bolt. In another application, a spur gear transfers power to other toothed surfaces, while a timing pulley transfers power to a timing belt. Both parts' shapes and functions are similar, but small details in tooth shape and design differentiate the two parts, as seen in Figure 3 (c). In contrast, changing the shape of common objects, like using longer legs on chairs, may not
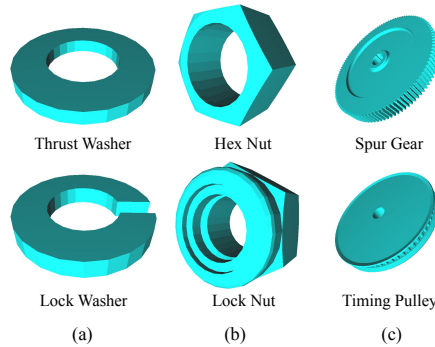
change the function of the object. Because these characteristics do not appear in common object datasets [4, 47], the existing shape descriptors [24, 36, 41] need to be benchmarked on MCB to explore the shape descriptors on mechanical components. This is because recently proposed deep neural networks descriptors are developed to capture the features from the common objects but not validated on the mechanical components. In this sense, Koch *et al.* [22] created a large CAD model dataset and benchmarked surface normal estimation, but they could not benchmarked object classification or shape retrieval because they are not labeled.

An annotated benchmark dataset such as MCB can link the shape to the particular representation inside product data management systems of CAD kernels. Our work opens up ways for implementation of fine-grained searches with features of mechanical components, semantic text, and mechanical meta-data.
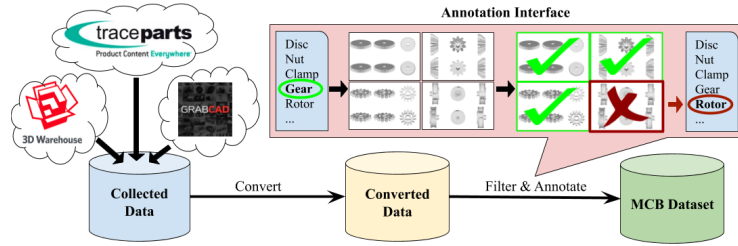

Fig. 4. Data acquisition and annotation overview for the creation of a large-scale mechanical components benchmark.

4   Dataset creation

For the dataset creation, we first elaborate on the acquisition of mechanical components and explain how we annotated them. We acquire models from online 3D CAD repositorys. To effectively annotate, CAD models are filtered and annotated using web-based tools. We define classes by following the field "Mechanical Systems and Components" of the International Classification Standard (ICS).

4.1   Data acquisition

We collect mechanical components from online large 3D CAD repositories: TraceParts[1], 3D Warehouse[2], and GrabCAD[3]. 3D Warehouse and GrabCAD are large online open repositories for professional designers, engineers, manufacturers, and students to share CAD models. They provide numerous CAD models

---

[1]https://www.traceparts.com/
[2]https://3dwarehouse.sketchup.com/
[3]https://grabcad.com/

with various classes, including mechanical components. The models from TraceParts are industry standard components and shape variation within class is small. By merging the models from different sources, we obtained 163K mechanical components before annotation and purification as shown in Table 2.

4.2 Acquired dataset purification

We developed a web-based platform to manage large-scale dataset functioning, collecting, viewing, filtering, and annotating data. The overview of the platform is available in Figure 4. Web-based applications have the advantage that users are free of installation and can easily access to the platform from any computer with internet connection. This accessibility accelerated the annotation process. We utilized the tool to trigger the scrapper collecting CAD models, also filtering and annotating the data with intuitive user interfaces, which is available in Figure 4. A dataset managing platform visualizes multi-view images of each engineering part, which gives users a more comprehensive understanding of the mechanical part during filtering and annotating. The data creation pipeline consists of consecutive three steps.

Step 1: *Conversion / Multi-view image generation.* The file format conversion process is necessary to create a unified file format dataset, since collected CAD models consist of various formats such as STL, STEP, and OFF. The converter module in the platform converts file format into OBJ format and captures projected images from multiple viewpoints. For 3D data conversion, we used Open Asset Import Library (Assimp) [1] and Gmsh [13]. We used projected images for annotating engineering parts.

Step 2: *Filtering.* We filter scrapped CAD models by deleting broken and duplicated models and capturing wrongly categorized models for re-annotation. Meta-information (i.e. file size, file name, search keyword, and data source) tagged in the process of scrapping step helps users to filter the data. Eight ME experts manually filtered out the duplicates with our annotation tool. We group objects with similar meta-information and these experts manually removed duplicates. An annotation interface presents the models in a table format rather than one model at a time. Several models can be viewed at one time, which increases the speed of filtering and making identifying duplicate models easier. A quantitative comparison of the dataset between before and after filtering is shown in Table 2.

Table 2. The number of data before and after filtering.

Data source	#Data	
	Before	After
GrabCAD	22,703	5,301
3D Warehouse	20,478	12,737
TraceParts	120,665	40,658
Total	163,216	58,696

Step 3: *Annotation.* After filtering, we re-annotate the missed categorized models to the correct category base on tagged information and a multi-view image. We use a 3D viewer in the annotation interface to present a close-up look when the multi-view image does not provide enough information to annotate for annotation. Some of the models do not belong to any of our mechanical

components categories but are still relevant to engineering parts. We defined these models as miscellaneous and label them into new part categories as needed.

5   Experiments

To analyze the behavior of learning algorithms developed for common objects works on mechanical components, we benchmarked classification and retrieval tasks with three different representations: point could, projected views, and voxel grids. We use two NVIDIA GeForce RTX 2080Ti GPUs, i9-9900k CPU, and 64GB RAM for the experiments. We carefully choose seven state-of-the-art shape classification algorithms from three different 3D shape representations: point cloud, multi-view, and voxel grids as the benchmark methods. In point cloud method, we use PointCNN [24], PointNet++ [36], and SpiderCNN [48]. For the multi-view based, we use MVCNN [44] and RotationNet [20]. DLAN [11] and VRN [3] are used to evaluate voxel grids representation. For training each method, we use the code and the hyper-parameters from seven deep-learning algorithm papers. We use 2,048 points density for point cloud, $32{\times}32{\times}32$ grid for voxel grids, and $3{\times}224{\times}224$ resolution for image-based representations. We follow the original papers for the input data processing and training procedures. For all the benchmark datasets, we randomly split the datasets into train and test set as 80% and 20%, respectively. Training is conducted for each method to prevent initialization variation and report the best results.

Point Clouds A point cloud is a collection of points in Euclidean space. PointCNN [24] relaxes irregularity of point clouds by approximating the transformation matrix with multi-layer perception, which simultaneously weights and permutes the input features for point cloud data feature learning. PointNet [35] learns a set of optimization functions for selecting feature points that contain meaningful content, which canonicalizes the input point clouds and aggregates all feature points to capture global point cloud features. PointNet++ [35] is an advanced version of PointNet. This work focused on recognizing fine-grained patterns with a hierarchical neural network which iteratively applied on a nested partitioned point set. SpiderCNN [48] proposes a convolutional layer, which is a product of a step function that captures local geodesic information and a Taylor polynomial to convolve in point cloud.

Projected Views View-based methods [20, 43, 44] extract features of 3D shape representations by observing multi-view images of an object and jointly estimating their poses. Their method successfully works for object classification and shape retrieval tasks, but performed poorly on unknown orientation models. Su *et al.* [43] uses a collection of multiple views of 3D objects, which is effective for learning their representations. MVCNN [44] further improves Su *et al.* with cross-modal distillation and adversarial inputs with a differentiable renderer.

Voxel Grids Three-dimensional objects can be discredited and represented in voxel grids, and voxel-based classifiers use voxel grids as their inputs. DLAN [11] proposes Rotation Normalized Grids (RNGs), which are samples of oriented point sets rotated by PCA for shape retrieval. Multiple blocks of RNGs are converted into local features with 3D convolution, and these features are aggregated with average pooling as object representations. VSL [27] learns the probabilistic manifold of the underlying structure of voxelized 3D shapes with an auto-encoder in an unsupervised manner. VoxNet [30] converts point clouds into voxels in voxel grids and extracts features with a 3D convolution layer for the classification tasks. VRN [3] uses a series of 3D convolutional layers to extract features for classifying objects compactly.

Table 3. The statistics of Mechanical Components Benchmark dataset.

Class	#Models	Class	#Models	Class	#Models
Articulations eyelets&joints	1,632	Impeller	145	Socket	858
Bearing accessories	107	Keys and keyways splines	4,936	Spacers	113
Bushes	764	Knob	644	Split pins	472
Cap nuts	225	Lever	1,032	Spring washers	55
Castle nuts	226	Locating pins	55	Springs	328
Castor	99	Locknuts	254	Square	72
Chain drives	100	Lockwashers	434	Square nuts	53
Clamps	155	Nozzle	154	Standard fitting	764
Collars	52	Plain guidings	49	Studs	4,089
Conventional rivets	3,806	Plates circulate plates	365	Switch	173
Convex washer	91	Plugs	169	T-nut	101
Cylindrical pins	1,895	Pulleys	121	T-shape fitting	338
Elbow fitting	383	Radial contact ball bearings	1,199	Taper pins	1,795
Eye screws	1,131	Right angular gearings	60	Tapping screws	2,182
Fan	213	Right spur gears	430	Threaded rods	1,022
Flange nut	53	Rivet nut	51	Thrust washers	2,333
Flanged block bearing	404	Roll pins	1,597	Toothed	47
Flanged plain bearings	110	Screws&bolts \w countersunk head	2,452	Turbine	85
Grooved pins	2,245	Screws&bolts \w cylindrical head	3,656	Valve	94
Helical geared motors	732	Screws&bolts \w hexagonal head	7,058	Washer bolt	912
Hexagonal nuts	1,039	Setscrew	1,334	Wheel	243
Hinge	54	Slotted nuts	78	Wingnuts	50
Hook	119	Snap rings	609	**Total**	**58,696**

5.1   Statistics of the dataset

MCB has a total number of 58,696 mechanical components with 68 classes. The exact name of the types and amount of data in each category are shown in Table 3. Objects from TraceParts are aligned, but the objects from the other two sources (30 % of the objects) are not consistently oriented. We did not perform additional alignments as many object classes do not possess consistent orientations due to a variety of continuous/discrete symmetries. On the other hand, having unaligned models in shape classification and retrieval tasks helps to evaluate the generalization of the shape descriptors [40]. Unlike 3D Warehouse and GrabCAD that provide data from general usages, TraceParts stores data from the manufacturing companies. The CAD models from manufacturing companies

show a tiny variation because they follow the parameterized catalogs for standardization. Therefore, to see the effect of data that has dense distribution and orientation invariance, we built two datasets for the experiment:

- Dataset A (MCB): Aggregated data from TraceParts[1], 3D Warehouse[2], and GrabCAD[3]
- Dataset B: Aggregated data from 3D Warehouse and GrabCAD.

The dataset A has the same statistic with the original MCB dataset, and the dataset B has 18,038 data with 25 classes. The detailed statistics of the dataset B is explained in supplementary material.

Table 4. Summary table of evaluation metrics of shape retrieval benchmark for seven deep learning methods. They are grouped by their representation types. Each *, †, and ⊠ symbol indicates the method point cloud, volumetric, and image, respectively.

| Dataset	Method	micro					macro				
		P@N	R@N	F1@N	mAP	NDCG@N	P@N	R@N	F1@N	mAP	DCG@N
A	PointCNN* [24]	**0.892**	**0.892**	**0.690**	**0.889**	**0.898**	0.869	**0.797**	**0.833**	**0.886**	**0.854**
	PointNet++* [36]	0.778	0.778	0.613	0.794	0.754	0.772	0.678	0.712	0.803	0.746
	SpiderCNN* [48]	0.839	0.839	0.669	0.867	0.793	0.844	0.741	0.776	0.877	0.812
	MVCNN⊠ [44]	0.579	0.579	0.488	0.657	0.487	0.667	0.552	0.585	0.735	0.641
	RotationNet⊠ [20]	0.688	0.699	0.508	0.805	0.683	0.784	0.652	0.683	0.815	0.735
	DLAN† [11]	0.840	0.840	0.568	0.879	0.828	**0.878**	0.786	0.820	0.880	0.845
	VRN† [3]	0.537	0537	0.402	0.653	0.519	0.646	0.480	0.507	0.664	0.576
B	PointCNN*	0.905	0.905	**0.676**	**0.913**	0.899	0.895	0.829	0.853	**0.909**	**0.871**
	PointNet++*	0.847	0.847	0.657	0.892	0.798	0.873	0.799	0.823	0.903	0.846
	SpiderCNN*	0.779	0.779	0.609	0.829	0.728	0.782	0.698	0.719	0.841	0.757
	MVCNN⊠	0.786	0.786	0.609	0.831	0.742	0793	0.719	0.741	0.852	0.776
	RotationNet⊠	0.529	0.529	0.434	0.607	0.454	0.560	0.466	0.483	0.647	0.540
	DLAN†	**0.912**	**0.912**	0.674	0.908	**0.925**	**0.903**	**0.830**	**0.854**	0.902	0.870
	VRN†	0.607	0.607	0.460	0.628	0.613	0.565	0.468	0.484	0.619	0.534
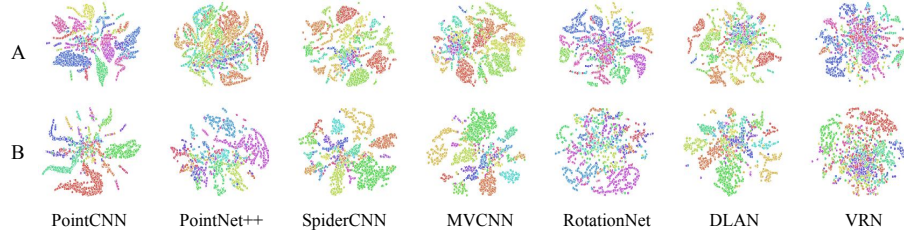
Fig. 5. t-SNE [29] plots of seven different deep neural networks trained with Dataset A and B. We set perplexity as 40 and iterate 300 times.
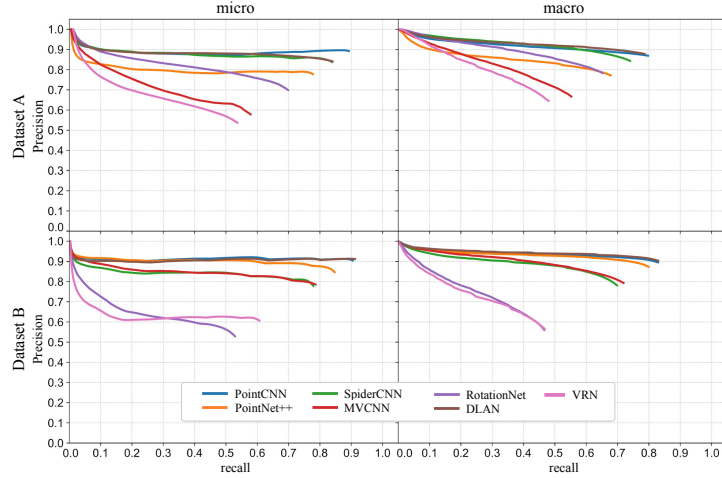
Fig. 6. Precision-recall curve plots for retrieval with seven different methods. The PointCNN shows best retrieval results for dataset A, and DLAN shows best retrieval results on dataset B.

5.2 Retrieval benchmark

At each entry, we calculate scores of the precision-recall curve in the retrieval results: precision, recall, F1-score, mAP, and Normalized Discounted Cumulative Gain (NDCG). In shape retrieval, NDCG has a heavier tail at high ranks, which means that it does not discount lower ranks as much as mAP does [41]. Therefore, NDCG has a better ability to show the ration between the real performance and ideal performance to evaluate the metrics.

Since each object has a different number of positive retrievals, the score table metrics are referred to as P@N, R@N, F1@N, and NDCG@N, where the N refers to the total retrieval list length of each object, which varies across queries. The macro-averaged version presents the performance of the dataset combining the result of each category. The micro-averaged version treats each query, and the retrieval result equally treats cross groups. Therefore, it eventually has the same P@N and R@N.

The summary results of all tested methods are given in Table 4. Corresponding precision-recall curves are given in Figure 6. To see the similarity of the geometric features from the descriptors, we perform t-distributed stochastic neighbor embedding (see Figure 5). We observe that the more the clusters are grouped, the more the retrieval results enhanced. Orientation invariance is crucial for the retrieval task. For example, DLAN and PointCNN, which have rotation invariance, perform best on both datasets. However, VRN and RotationNet show poor results on dataset B where the orientation is not aligned, even though it uses the same representation as DLAN. RotationNet also poorly performed on the common shape retrieval task [40] when the orientations of the objects are perturbed.

The overall retrieval performance of dataset A is relatively higher than dataset B. Micro has slightly better results on P@N, while much better results on R@N show that the metrics have better performance in cross-category testing.

We observe that the performance of RotationNet and VRN dramatically decreases on dataset B compared to dataset A. This is because the object orientations are aligned on dataset A but not on B. Similar behavior is observed for the common objects [41]. Specifically, RotationNet predicts view orders of given multi-views to learn rotation-invariant features. However, the camera viewpoints of solid of revolution shapes given multi-views are hard to determine and impossible to predict when the cameras are rotating along with the center axis of the object. DLAN and PointCNN perform well on both datasets, with respect to both macro and micro metrics. We conclude that these methods extract rotation-invariant features across classes. As a point of view in data representation, point cloud methods show stable performance on both datasets.
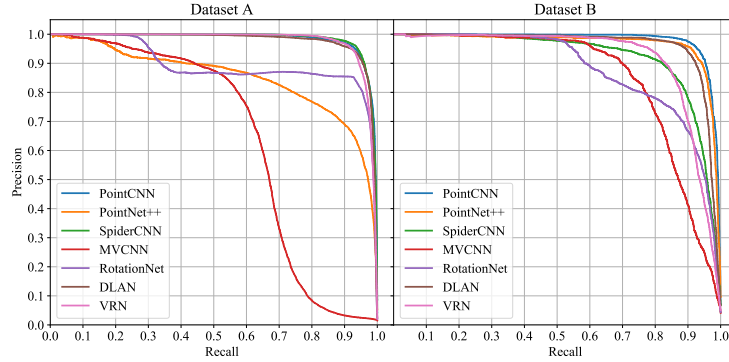
Fig. 7. Precision and Recall curve plots of classification task. Left plot shows the PR curve of dataset A and right plot shows the PR curve dataset B. The RotationNet shows the best performance in terms of accuracy.

5.3   Classification benchmark

For the classification task, we measure four metrics, mean accuracy over objects, average accuracy per class, F1-score and Average Precision (AP) and plotted precision-recall curves. We use the macro method for F1 and AP calculation. AP metrics are used to compare the network performance across the dataset A and B. F1 score is harmonic mean of the precision and recall. The benchmark result on Dataset A and B are available in Table 5 and Figure 7. Additionally, to compare the performance between common objects and mechanical objects, we provide classification performance on MondelNet40 on Table 6.

Table 5. Benchmark results of the seven classification models which were trained and evaluated on our mechanical engineering part benchmark. We trained five times per model and reported the highest result. Each *, †, and ⊠ symbol indicates the method: point cloud, volumetric, and image representation, respectively.

Method	Acc. over object (%)		Acc. over class (%)		F1-score		Average Precision	
	A	B	A	B	A	B	A	B
PointCNN* [24]	93.89	93.67	81.85	86.80	83.86	88.63	**90.13**	**93.86**
PointNet++* [36]	87.45	93.91	73.68	87.97	74.59	88.32	73.45	91.33
SpiderCNN* [48]	93.59	89.31	79.70	79.29	81.30	80.72	86.64	82.47
MVCNN⊠ [44]	64.67	79.17	80.47	84.09	69.69	77.69	79.82	86.66
RotationNet⊠ [20]	**97.35**	**94.73**	**90.79**	**89.70**	**92.29**	**91.05**	87.58	84.87
DLAN† [11]	93.53	91.38	82.97	84.21	83.81	83.88	89.80	90.14
VRN† [3]	93.17	85.44	80.34	70.15	81.48	73.01	85.72	77.36

Unlike the retrieval task, RotationNet outperforms the other methods on both datasets (see Table 5 and 6). The performance of MVCNN drops significantly on the mechanical components compared to the common objects which is ModelNet40. On the other hand, the accuracy of RotationNet drops slightly. The major differences between MVCNN and RotationNet are estimating correspondence between each image and view order during training. This correspon-

Table 6. Classification accuracy on Model-Net40. Each *, †, and ⊠ symbol indicates the method is based on point cloud representation, volumetric representation, and images representation, respectively.

Method	Acc. over object (%)
PointCNN* [24]	92.2
PointNet++* [36]	91.9
SpiderCNN* [48]	92.4
MVCNN⊠ [44]	95.0
RotationNet⊠ [20]	**97.37**
DLAN† [11]	84.0
VRN† [3]	95.5

dence estimation relaxes rotation variant property by implicitly learning mapping function between each view of the object and camera view point. In point cloud methods, PointCNN shows the best performance on both datasets, and SpiderCNN perform better on Dataset A than B. PointCNN performs best for the AP (see Table 5). This is because mechanical components are sensitive to the local changes and PointCNN leverage spatially-local correlation. In the same sense, DLAN perform better on mechanical components due to oriented point sets. However, VRN performance drops on the mechanical components benchmark since voxel grids are orientation variant.

From our benchmark result, capturing local features and having orientation invariance are crucial for developing mechanical components classifier. Although RotationNet shows 97 % accuracy over an object, accuracy over the class, which is 90.79%, is not good enough to utilize in the industry. For the deep learning application for mechanical components in the industry, a deeper understanding of mechanical parts is required. The classification result of the 'Flanged plain bearings' class shows a low accuracy, which is under 87% for every network. This value is relatively lower than the accuracy of other classes (see appendix).

This result shows the limitation of existing 3D object classification algorithms in terms of extracting local features. The general shape of the bearing is almost similar to thick washers or rings. Therefore, if the network cannot capture the local difference of ring-shaped object, it is hard to distinguish those objects.

We experiment how point cloud density affects the results in point cloud base algorithms. We perform five different density: 128, 256, 512, 1,024, and 2,048 points on three point cloud classification methods [24, 36, 48] for the classification task on Dataset B. From our experiment results, the performance increases as the density of the point cloud increases, as shown in line plots in Figure 8. However, the enhancement of results saturates as the point cloud density grows and the performance of Spi-

Fig. 8. Classification results of five different point cloud densities.

derCNN [48] decreases even the density increases from 1,025 to 2,048. Point-Net++ [36] is the most sensitive in the density of the point cloud, and PointCNN [24] is the least vulnerable in the variation of the point cloud density.

6   Conclusion

We propose a large-scale mechanical component benchmark with annotations. For the creation of the dataset, we develop an annotation framework that enhances the efficacy of the annotation and filtering processes. We perform shape classification and retrieval experiments with seven deep-learning shape classification methods which are designed to classify common objects. We find that view-based and voxel grid presentation-based methods perform poorly on random orientation of mechanical components. However, DLAN, a voxel-based method, performs well on random orientation since it has orientation invariance. The creation of MCB and experimental results can be used for the development of data-driven algorithms of mechanical components. For the future work, the development on dedicated data-driven feature descriptor for mechanical components will be conducted.

References

1. Open asset import library. `http://www.assimp.org/`, accessed: 2019-11-10
2. Bespalov, D., Ip, C.Y., Regli, W.C., Shaffer, J.: Benchmarking cad search techniques. In: Proceedings of the 2005 ACM symposium on Solid and physical modeling. pp. 275–286. ACM (2005)
3. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint arXiv:1608.04236 (2016)
4. Chang, A.X., Funkhouser, T., Guibas, L., Hanrahan, P., Huang, Q., Li, Z., Savarese, S., Savva, M., Song, S., Su, H., Xiao, J., Yi, L., Yu, F.: ShapeNet: An Information-Rich 3D Model Repository. Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago (2015)
5. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018)
6. Choi, C., Kim, S., Ramani, K.: Learning hand articulations by hallucinating heat distribution. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3104–3113 (2017)
7. Choy, C.B., Xu, D., Gwak, J., Chen, K., Savarese, S.: 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In: European conference on computer vision. pp. 628–644. Springer (2016)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
9. Ellis, K., Ritchie, D., Solar-Lezama, A., Tenenbaum, J.: Learning to infer graphics programs from hand-drawn images. In: Advances in neural information processing systems. pp. 6059–6068 (2018)
10. Farag, A., Ali, A., Graham, J., Farag, A., Elshazly, S., Falk, R.: Evaluation of geometric feature descriptors for detection and classification of lung nodules in low dose ct scans of the chest. In: 2011 IEEE International Symposium on Biomedical Imaging: from nano to macro. pp. 169–172. IEEE (2011)
11. Furuya, T., Ohbuchi, R.: Diffusion-on-manifold aggregation of local features for shape-based 3d model retrieval. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. pp. 171–178. ACM (2015)
12. Furuya, T., Ohbuchi, R.: Deep aggregation of local 3d geometric features for 3d model retrieval. In: BMVC. pp. 121–1 (2016)
13. Geuzaine, C., Remacle, J.F.: Gmsh: A 3-d finite element mesh generator with built-in pre-and post-processing facilities. International journal for numerical methods in engineering **79**(11), 1309–1331 (2009)
14. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
15. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017)
16. Huang, J., Kwok, T.H., Zhou, C.: Parametric design for human body modeling by wireframe-assisted deep learning. Computer-Aided Design **108**, 19–29 (2019)
17. Iyer, N., Jayanti, S., Lou, K., Kalyanaraman, Y., Ramani, K.: Three-dimensional shape searching: state-of-the-art review and future trends. Computer-Aided Design **37**(5), 509–530 (2005)

18. Jayanti, S., Kalyanaraman, Y., Iyer, N., Ramani, K.: Developing an engineering shape benchmark for cad models. Computer-Aided Design **38**(9), 939–953 (2006)
19. Jia, X., Gavves, E., Fernando, B., Tuytelaars, T.: Guiding the long-short term memory model for image caption generation. In: The IEEE International Conference on Computer Vision (ICCV) (December 2015)
20. Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5010–5019 (2018)
21. Kim, S., Winovich, N., Chi, H.G., Lin, G., Ramani, K.: Latent transformations neural network for object view synthesis. The Visual Computer pp. 1–15 (2019)
22. Koch, S., Matveev, A., Jiang, Z., Williams, F., Artemov, A., Burnaev, E., Alexa, M., Zorin, D., Panozzo, D.: Abc: A big cad model dataset for geometric deep learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9601–9611 (2019)
23. Kulkarni, T.D., Whitney, W.F., Kohli, P., Tenenbaum, J.: Deep convolutional inverse graphics network. In: Advances in neural information processing systems. pp. 2539–2547 (2015)
24. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. In: Advances in Neural Information Processing Systems. pp. 820–830 (2018)
25. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2117–2125 (2017)
26. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European conference on computer vision. pp. 740–755. Springer (2014)
27. Liu, S., Giles, L., Ororbia, A.: Learning a hierarchical latent-variable model of 3d shapes. In: 2018 International Conference on 3D Vision (3DV). pp. 542–551. IEEE (2018)
28. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: European conference on computer vision. pp. 21–37. Springer (2016)
29. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
30. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 922–928. IEEE (2015)
31. Miller, G.A.: WordNet: An electronic lexical database. MIT press (1998)
32. Mo, K., Zhu, S., Chang, A.X., Yi, L., Tripathi, S., Guibas, L.J., Su, H.: Part-net: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 909–918 (2019)
33. Monti, F., Boscaini, D., Masci, J., Rodola, E., Svoboda, J., Bronstein, M.M.: Geometric deep learning on graphs and manifolds using mixture model cnns. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
34. Paszke, A., Chaurasia, A., Kim, S., Culurciello, E.: Enet: A deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147 (2016)

35. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 652–660 (2017)
36. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017)
37. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 779–788 (2016)
38. Regli, W.C., Foster, C., Hayes, E., Ip, C.Y., McWherter, D., Peabody, M., Shapirsteyn, Y., Zaychik, V.: National design repository project: a status report. In: International Joint Conferences on Artificial Intelligence (IJCAI), Seattle, WA, Aug. pp. 4–10 (2001)
39. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241. Springer (2015)
40. Savva, M., Yu, F., Su, H., Aono, M., Chen, B., Cohen-Or, D., Deng, W., Su, H., Bai, S., Bai, X., et al.: Shrec16 track: largescale 3d shape retrieval from shapenet core55. In: Proceedings of the eurographics workshop on 3D object retrieval. pp. 89–98 (2016)
41. Savva, M., Yu, F., Su, H., Kanezaki, A., Furuya, T., Ohbuchi, R., Zhou, Z., Yu, R., Bai, S., Bai, X., et al.: Large-scale 3d shape retrieval from shapenet core55: Shrec'17 track. In: Proceedings of the Workshop on 3D Object Retrieval. pp. 39–50. Eurographics Association (2017)
42. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The princeton shape benchmark. In: Proceedings Shape Modeling Applications, 2004. pp. 167–178. IEEE (2004)
43. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 945–953 (2015)
44. Su, J.C., Gadelha, M., Wang, R., Maji, S.: A deeper look at 3d shape classifiers. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
45. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Thirty-First AAAI Conference on Artificial Intelligence (2017)
46. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3156–3164 (2015)
47. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
48. Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y.: Spidercnn: Deep learning on point sets with parameterized convolutional filters. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 87–102 (2018)

Appendix of
A Large-scale Annotated Mechanical Components Benchmark for Classification and Retrieval Tasks with Deep Neural Networks

Sangpil Kim[*1], Hyung-gun Chi[*1], Xiao Hu[1], Qixing Huang[2], and Karthik Ramani[1]

[1] Purdue University
[2] The University of Texas at Austin
{kim2030, chi45, hu440, ramani}@purdue.edu
huangqx@cs.utexas.edu

1   Statistics of dataset B

Dataset B is aggregated 3D mechanical components from 3D Warehouse[***] and GrabCAD[†]. The labels of dataset B do not follow the hierarchical taxonomy of MCB dataset. When creating the dataset A, we re-annotated and re-arranged each class of dataset B to merge with the data from TraceParts[‡], based on International Classification for Standards (ICS [1]). dataset B has total number of 18,038 models with 25 classes. A detailed statistics of the dataset is listed in Table 1.

Table 1. The statistics of dataset B.

Class	# Models	Class	# Models
Bearings	1,117	Nuts	1,069
Bushes	592	Pins	2,659
Castors	1,109	Plates	366
Clamps	157	Pullies	312
Discs	109	Rings	551
Fittings	1,756	Rivets	51
Flanges	398	Rotors	470
Fork joints	47	Screws	3,661
Gears	515	Springs	348
Handles	1,751	Studs	352
Hinges	61	Switches	177
Hooks	122	Washers	880
Motors	746	**Total**	**18,038**

---

[*] These authors made an equal contribution.
[***] https://3dwarehouse.sketchup.com/
[†] https://grabcad.com/
[‡] https://www.traceparts.com/

2   Experiments on only aligned data

We built Dataset C as aggregated data from only TraceParts which are all aligned. Then, we benchmarked classification and retrieval tasks with seven networks.

| PointCNN | PointNet++ | SpiderCNN | MVCNN | RotationNet | DLAN | VRN |
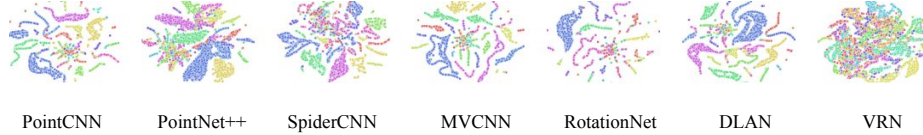
Fig. 1. T-SNE [6] plots of seven different deep neural networks trained with Dataset C. We set perplexity as 40 and iterate 300 times.

2.1   Retrieval benchmark

Table 2. Summary table of evaluation metrics of shape retrieval benchmark for seven deep learning methods. They are grouped by their representation types. Each *, †, and ⊠ symbol indicates the method point cloud, volumetric, and image, respectively.

Method	micro					macro				
	P@N	R@N	F1@N	mAP	NDCG@N	P@N	R@N	F1@N	mAP	DCG@N
PointCNN* [5]	0.912	0.912	0.689	0.901	0.923	0.860	0.802	0.826	0.880	0.845
PointNet++* [7]	0.870	0.870	0.666	0.872	0.862	0.834	0.773	0.796	0.862	0.817
SpiderCNN* [9]	0.908	0.908	0.703	0.922	0.881	0.883	0.809	0.833	0.906	0.857
MVCNN⊠ [8]	0.516	0.516	0.431	0.585	0.436	0.547	0.481	0.498	0.635	0.545
RotationNet⊠ [4]	0.756	0.756	0.63	0.836	0.618	0.813	0.701	0.722	0.851	0.760
DLAN† [3]	**0.952**	**0.952**	**0.719**	**0.941**	**0.961**	**0.919**	**0.866**	**0.891**	**0.927**	**0.903**
VRN† [2]	0.755	0.755	0.588	0.777	0.716	0.708	0.61	0.63	0.771	0.676
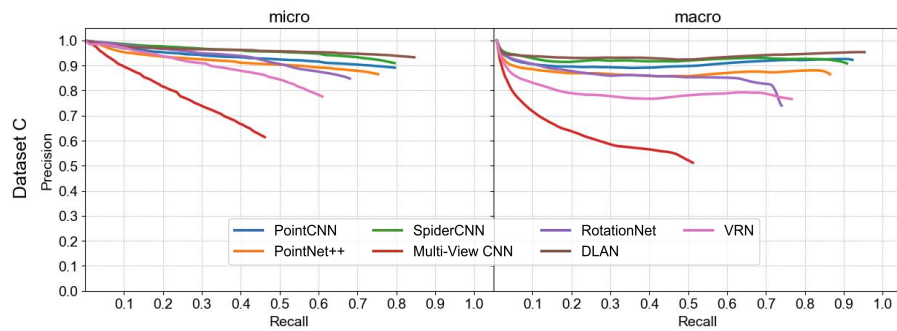
Fig. 2. Precision-recall curve plots for retrieval benchmark with seven different methods. The DLAN shows the best retrieval results.

2.2   Classification benchmark
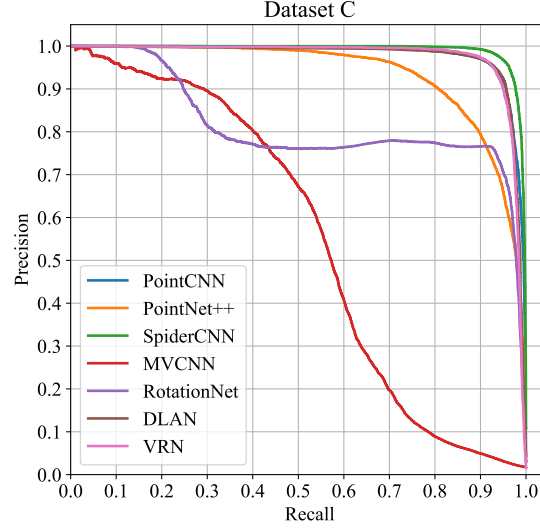
Fig. 3. Precision and Recall curve plots of classification task. The RotationNet shows the best performance in terms of accuracy.

Table 3. Benchmark results of the seven classification models which were trained and evaluated on our mechanical engineering part benchmark. We trained five times per model and reported the highest result. Each *, †, and ⊠ symbol indicates the method: point cloud, volumetric, and image representation, respectively.

Method	Acc. over object (%)	Acc. over class (%)	F1-score	Average Precision
PointCNN* [5]	93.77	71.91	74.46	83.70
PointNet++* [7]	89.97	78.82	78.48	79.45
SpiderCNN* [9]	95.68	80.97	82.12	88.78
MVCNN⊠ [8]	61.29	76.71	62.20	69.38
RotationNet⊠ [4]	96.08	82.61	84.75	80.47
DLAN† [3]	93.99	83.82	83.43	87.17
VRN† [2]	93.47	74.37	74.35	80.68

3   Density Distribution of Classification Results

We plot the probability density distribution of classification results of the Dataset A to clarify the inconsistency between PR curve and accuracy. In the Figure 4, 5, positive states the samples of one class, and all other samples are assigned to negative. Density distributions of positive and negative in RotationNet [4](Figure 4) are separated but closer than VRN [2](Figure 5) which explains the dropping of PR curve of classification task. Other networks that show high classification accuracy and a good PR curve have similar density distribution as VRN.
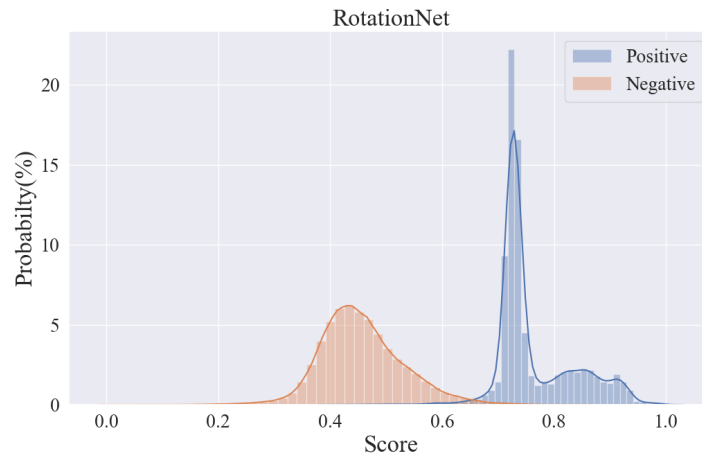
Fig. 4. Probability density distribution of classification results of RotationNet.
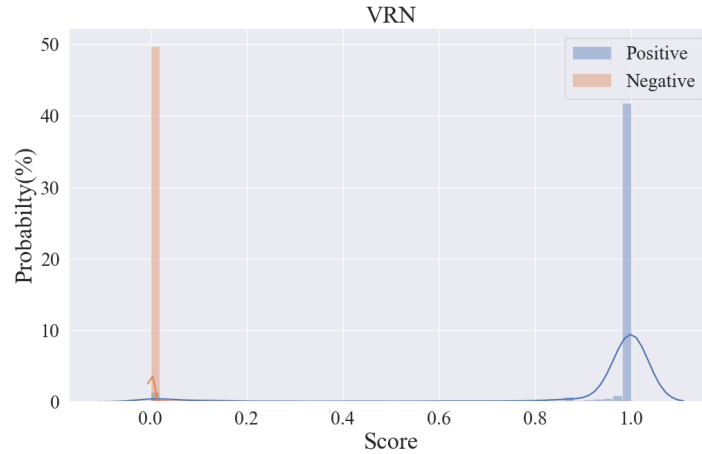
Fig. 5. Probability density distribution of classification results from VRN.

4   Detail information of the retrieval benchmark

Figure 7 shows the hierarchical taxonomy of Mechanical Components Benchmark (MCB). We provide confusion matrix plot of seven methods [2–5,7–9] to visualize the details of classification results on Figure 8 and 9. The accuracy of individual class of each method on both datasets A and B are listed in Table 4 and 5. The representative mechanical objects are shown in Figure 10. The retrieval results are depicted in Figure 6.
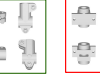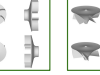
Fig. 6. Sample retrieval results of category Clamp and Rotor. N is the number of data in the corresponding class of the query. Within the first N retrievals, green-border images represent the true positive results, and red-border images represent the false positive results.
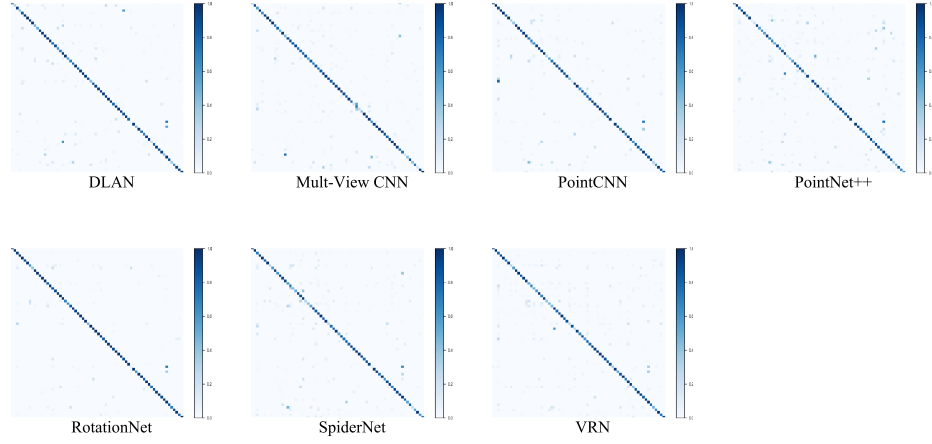
Fig. 7. The hierarchy taxonomy of mechanical components which are based on the International Classification for Standards (ICS).

DLAN   Mult-View CNN   PointCNN   PointNet++

RotationNet   SpiderNet   VRN

Fig. 8. Confusion Matrix of seven different deep neural networks trained with the dataset A.

DLAN   Multi-View CNN   PointCNN   PointNet++

RotationNet   SpiderNet   VRN

Fig. 9. Confusion Matrix of seven different deep neural networks trained with the dataset B.

Fig. 10. Samples of Mechanical Components Benchmark (MCB) dataset of a total 68 classes.

Table 4. Classification results of dataset A for seven different benchmark and 68 classes.

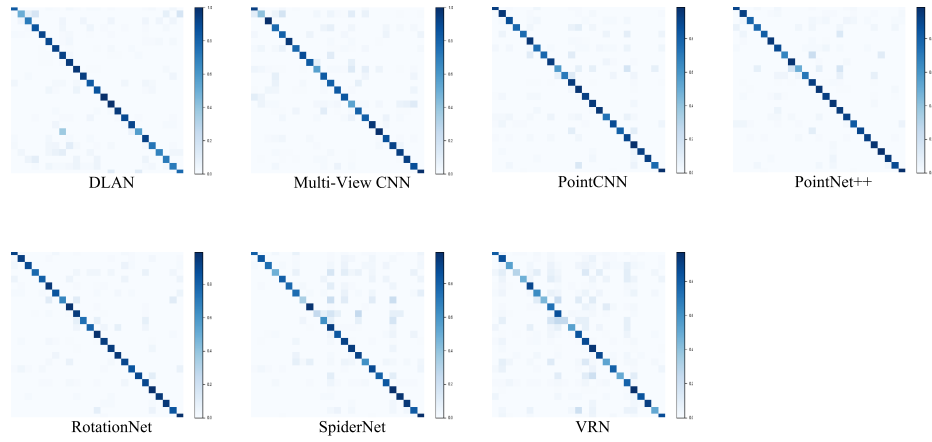| Class	Point CNN	Point Net++	Spider Net	Multi-View CNN	Rota-tion Net	DLAN	VRN
Articulations eyelets & joints	0.988	0.880	1.000	0.452	1.000	0.964	0.997
Bearing accessories	1.000	1.000	1.000	0.917	1.000	0.333	1.000
Bushes	0.914	0.776	0.722	0.580	0.947	0.929	0.789
Cap nuts	0.978	0.911	0.933	1.000	1.000	0.429	0.911
Castle nuts	1.000	1.000	1.000	0.618	1.000	0.862	1.000
Castor	0.789	0.632	0.789	0.978	0.895	1.000	0.737
Chain drives	1.000	0.950	1.000	1.000	1.000	1.000	0.950
Clamps	0.355	0.097	0.290	0.947	0.645	0.947	0.323
Collars	0.400	0.200	0.600	1.000	0.400	1.000	0.600
Conventional rivets	0.993	0.974	1.000	0.710	0.999	0.484	0.986
Convex washer	1.000	0.556	0.611	0.800	1.000	0.500	1.000
Cylindrical pins	0.813	0.633	0.960	0.543	0.908	0.993	0.863
Elbow fitting	0.961	0.566	0.920	1.000	1.000	0.944	0.974
Eye screws	0.965	0.947	0.942	0.449	0.987	0.953	0.960
Fan	0.738	0.667	0.619	0.921	0.833	0.987	0.548
Flanged block bearing	0.688	0.688	0.734	0.912	1.000	0.969	0.700
Flanged plain bearings	0.636	0.500	0.455	0.857	0.875	0.762	0.488
Flange nut	0.900	0.900	0.500	0.900	0.909	0.900	0.591
Grooved pins	0.967	0.644	0.967	0.788	0.996	0.625	1.000
Helical geared motors	0.938	0.808	0.903	0.773	0.986	0.682	0.699
Hexagonal nuts	0.981	0.942	0.952	0.808	0.986	0.966	0.971
Hinge	0.300	0.300	0.300	0.889	0.700	0.981	0.400
Hook	0.565	0.435	0.435	0.600	0.652	0.600	0.478
Impeller	0.724	0.552	0.536	0.826	0.828	0.696	0.448
Keys and keyways, splines	1.000	0.991	1.000	0.750	1.000	0.759	1.000
Knob	0.847	0.756	0.817	0.848	0.931	1.000	0.771
Lever	0.966	0.922	0.946	0.771	0.990	0.824	0.888
Locating pins	0.909	0.727	1.000	0.845	1.000	0.961	1.000
Locknuts	0.860	0.280	0.840	1.000	0.940	1.000	0.860
Lockwashers	0.895	0.953	0.977	0.960	1.000	0.980	0.884
Nozzle	0.467	0.067	0.333	0.953	0.500	0.919	0.300
Plain guidings	0.111	0.889	0.778	0.833	1.000	0.433	0.889
Plates, circulate plates	0.973	0.877	0.822	1.000	0.986	0.889	0.288
Plugs	1.000	0.970	1.000	0.918	0.970	0.945	0.939
Pulleys	0.583	0.458	0.542	1.000	1.000	0.970	0.667
Radial contact ball bearings	0.967	0.967	0.954	1.000	0.987	0.958	0.950
Right angular gearings	0.750	0.500	0.750	0.891	0.833	0.987	0.583
Right spur gears	0.886	0.852	0.862	0.830	1.000	0.932	0.841
Rivet nut	0.800	0.700	0.800	1.000	1.000	1.000	0.800
Roll pins	0.997	0.975	0.997	0.799	0.997	0.981	0.984
Screws and bolts with countersunk head	0.886	0.716	0.880	0.182	0.908	0.804	0.941
Screws and bolts with cylindrical head	0.985	0.877	0.989	0.720	0.995	0.995	0.970
Screws and bolts with hexagonal head	0.999	0.984	0.990	0.266	0.998	0.980	0.979
Setscrew	0.981	0.917	0.989	0.436	0.992	0.992	0.985
Slotted nuts	0.800	0.800	0.800	1.000	0.867	0.867	0.867
Snap rings	0.950	0.983	0.926	0.950	1.000	0.992	0.909
Socket	0.953	0.924	0.930	0.959	0.971	0.784	0.924
Spacers	0.000	0.000	0.091	0.682	0.091	0.045	0.409
Split pins	0.968	0.979	0.968	1.000	0.979	0.979	0.979
Springs	0.938	0.954	0.969	1.000	0.636	0.455	0.545
Spring washers	0.636	0.818	0.727	0.969	1.000	0.985	0.862
Square	1.000	0.714	1.000	1.000	1.000	0.786	1.000

Square nuts	0.700	0.700	0.600	0.900	0.800	0.700	0.700
Standard fitting	0.947	0.875	0.887	0.803	0.993	0.954	0.967
Studs	0.985	0.935	0.987	0.962	0.996	0.968	0.994
Switch	0.618	0.588	0.706	0.971	0.765	0.235	0.412
Taper pins	0.875	0.958	0.964	1.000	0.850	0.800	0.850
Tapping screws	0.823	0.826	0.906	0.940	0.970	0.970	0.970
Threaded rods	0.760	0.721	0.971	0.735	1.000	0.880	0.972
Thrust washers	0.989	0.961	0.989	0.796	0.911	0.961	0.814
T-nut	0.850	0.700	0.800	0.196	1.000	0.475	0.961
Toothed	0.889	0.889	0.556	0.416	0.996	0.996	0.994
T-shape fitting	0.925	0.851	0.836	0.889	0.889	0.889	0.778
Turbine	0.294	0.353	0.235	0.412	0.706	0.353	0.471
Valve	0.611	0.500	0.556	0.778	0.944	0.667	0.667
Washer bolt	0.978	0.852	0.989	0.159	0.989	0.951	0.989
Wheel	0.813	0.688	0.604	0.938	0.813	0.917	0.771
Wingnuts	0.900	0.600	0.700	1.000	1.000	1.000	0.900
Bearings	0.923	0.885	0.808	0.946	0.923	0.904	0.808
Bushes	0.966	0.915	0.839	0.403	0.941	0.737	0.856

Table 5. Classification results of dataset B for seven different benchmark and 25 classes.

Class	Point CNN	Point Net++	Spider Net	Multi-View CNN	Rotation Net	DLAN	VRN
Castors	0.875	0.819	0.750	1.000	0.889	0.931	0.542
Clamps	0.806	0.806	0.484	0.758	0.871	0.677	0.290
Discs	0.762	0.762	0.810	0.810	0.810	0.714	0.476
Fittings	0.963	0.957	0.778	0.967	0.957	0.960	0.883
Flanges	0.747	0.911	0.769	0.785	0.861	0.709	0.633
Fork joints	0.667	0.667	0.333	0.889	0.667	0.778	0.444
Gears	0.922	0.961	0.971	0.833	0.990	0.902	0.725
Handles	0.937	0.943	0.929	0.558	0.946	0.914	0.857
Hinges	0.667	0.583	0.250	0.875	0.833	0.500	0.250
Hooks	0.708	0.667	0.625	0.812	0.792	0.708	0.542
Motors	0.973	0.973	0.839	0.859	0.980	0.933	0.839
Nuts	0.930	0.925	0.916	0.850	0.893	0.841	0.869
Pins	0.951	0.970	0.943	0.560	0.955	0.930	0.945
Plates	0.945	0.945	0.918	0.822	0.959	0.945	0.877
Pullies	0.754	0.918	0.617	0.792	0.902	0.836	0.525
Rings	0.900	0.827	0.890	0.987	0.845	0.864	0.791
Rivets	0.800	0.900	0.800	1.000	0.900	1.000	0.500
Rotors	0.892	0.935	0.839	0.889	0.892	0.86	0.785
Screws	0.985	0.981	0.985	0.911	0.992	0.996	0.971
Springs	0.942	0.986	0.971	0.949	0.971	0.957	0.841
Studs	0.929	0.943	0.957	0.943	0.971	0.914	0.914
Switches	0.771	0.886	0.829	0.871	0.914	0.571	0.514
Washers	0.983	0.989	0.989	1.000	0.989	0.971	0.863

References

1. International Classification for Standards. ISO, 7 edn. (2015)
2. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint arXiv:1608.04236 (2016)
3. Furuya, T., Ohbuchi, R.: Diffusion-on-manifold aggregation of local features for shape-based 3d model retrieval. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval. pp. 171–178. ACM (2015)
4. Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5010–5019 (2018)
5. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. In: Advances in Neural Information Processing Systems. pp. 820–830 (2018)
6. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. Journal of machine learning research **9**(Nov), 2579–2605 (2008)
7. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In: Advances in neural information processing systems. pp. 5099–5108 (2017)
8. Su, J.C., Gadelha, M., Wang, R., Maji, S.: A deeper look at 3d shape classifiers. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 0–0 (2018)
9. Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y.: Spidercnn: Deep learning on point sets with parameterized convolutional filters. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 87–102 (2018)