A Large-scale Mechanical Components Benchmark for Deep Neural Networks

Anonymous CVPR submission

Paper ID 1858

Abstract

We introduce a large-scale mechanical components benchmark for the classification and retrieval tasks named Mechanical Components Benchmark (MCB): a large-scale dataset of 3D objects of mechanical components. The dataset enables the data-driven feature learning for the mechanical components. Exploring the descriptor for mechanical components is essential to the computer vision, mechanical design and manufacturing applications. However, limited works have been done for creating the annotated mechanical components dataset on a large-scale. This is because annotating mechanical components require engineering knowledge, and acquiring a 3D model is challenging. With our annotated dataset, we 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. We further evaluated the features representation of each trained classifier by performing mechanical components retrieval to examine the behavior of each method on mechanical components. The main contributions are the creation of a large-scale annotated mechanical component benchmark, defining hierarchy taxonomy of mechanical components, and benchmark the effectiveness of deep learning shape classifiers on the mechanical components.

1. Introduction

The application of machine learning has been highlighted recently due to the effectiveness of the deep neural networks [34, 11, 29, 33, 15, 19]. Alone with deep neural networks, data driven algorithm and the creation of a large-scale dataset [8, 4, 43, 24] have led to a series of breakthroughs in computer vision [41, 13, 6] and graphics [21, 9, 30]. As shown in the ImageNet [8] dataset which followed the hierarchical structure of WordNet [28], a well-structured and annotated dataset can be used for training feature extractor. The pretrained feature extractor from the ImageNet has been widely used to generates feature representations that are used in object detection [34, 26, 23], image segmentation [31, 5, 14, 36], image caption generator [42, 17], and image retrieval [18, 39].

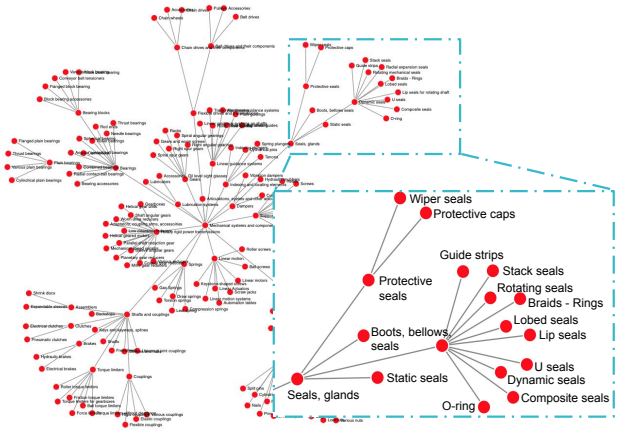

Figure 1: The hierarchy taxonomy of mechanical components which are based on the International Classification for Standards (ICS).

However, in the field of mechanical design and manufacturing applications, 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, no direct comparison is available for competing methods to improve the learning algorithms [38]. The creation of large-scale mechanical components dataset with a well-organized class hierarchy and annotations is crucial in the computer vision, manufacturing industry, and mechanical communities. Besides inspiring creation of knowledge network, categorized mechanical components dataset is also vital to mechanical Product Lifecycle Management (PLM) and industrial simulation. In addition, our pre-annotated database can help computer vision research field since tiny features in similar shapes lead to different functionalities.

Creating a large-scale mechanical component dataset is challenging due to the significant difficulty in collecting 3D CAD models of mechanical components. Different from

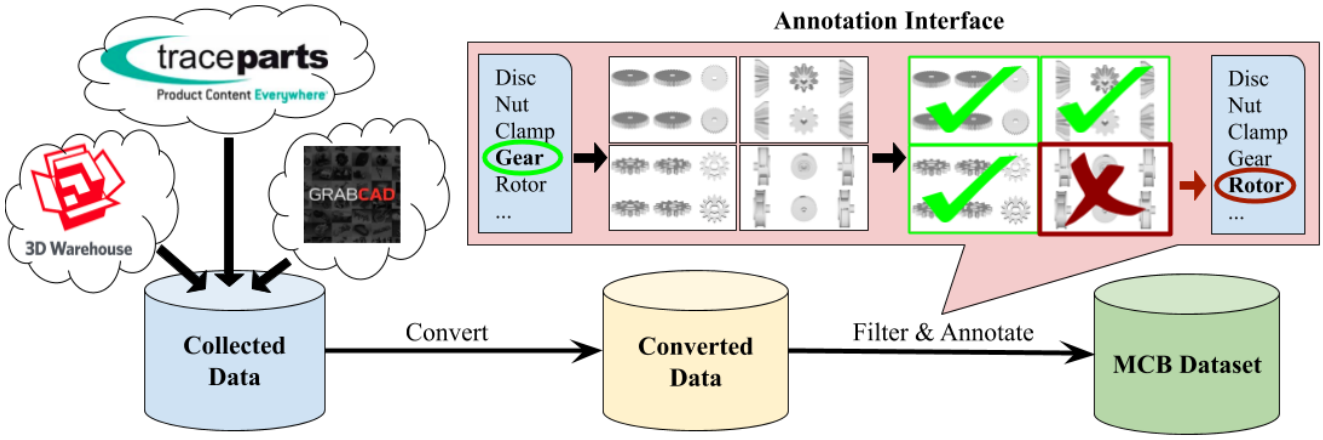

Figure 2: Data acquisition and annotation overview. We created a large-scale mechanical components benchmark.

creating common-object datasets [4, 43, 38, 29], the accessibility of most mechanical components is limited because of proprietary and ownership issues of models involving design knowledge. Product and manufacturing models are held by companies for commercial usages, resulting in a deficiency of open-source components datasets. Additionally, the inconsistency and incompatibility of mechanical components from available sources require massive effort on filtering and annotating the data. Also, annotating mechanical components is harder than annotating common objects. It is highly demanding on annotators' knowledge and expertise 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 International Organization for Standardization (ISO), the tree model is depicted in Figure 1. We developed a web-based application, which is shown in Figure 2 to reduce the difficulty of filtering and annotating. The application supports user-controllable viewing, displaying meta information, annotating, and filtering parts by viewing multiple parts, each part as a collection of six projected views, as a tabular. These functionalities make the benchmark creation faster and more accurate for engineering parts with similar structures. Also, instruction includes meta-information of mechanical components guiding annotators to remove duplicated parts.

Furthermore, we benchmark seven state-of-the-art shape descriptors to analyze the properties of mechanical components. Seven methods were carefully selected from three different 3D object representations: (1) point clouds, (2) voxel girds, and (3) view-based. We observed that view-based methods are weak on unseen orientation in shape retrieval tasks, which is also observed in common object retrieval tasks [37]. However, some methods showed different results from common objects 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, and accuracy per class, accuracy per instance, F1 score, average precision for classification tasks.

Our main contributions:

- We create a hierarchical taxonomy based on both the functionality and shape of mechanical components and develop the web-based framework, which has a viewing, annotating, and filtering feature to guide annotator.
- We annotate and collect a large-scale mechanical components benchmark. The mechanical components are annotated by mechanical expertise based on their functionality and shapes.
- We benchmark seven deep learning-based classifiers and analyze their performances with the mechanical components dataset.

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

2. Related works

We 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 5.2.

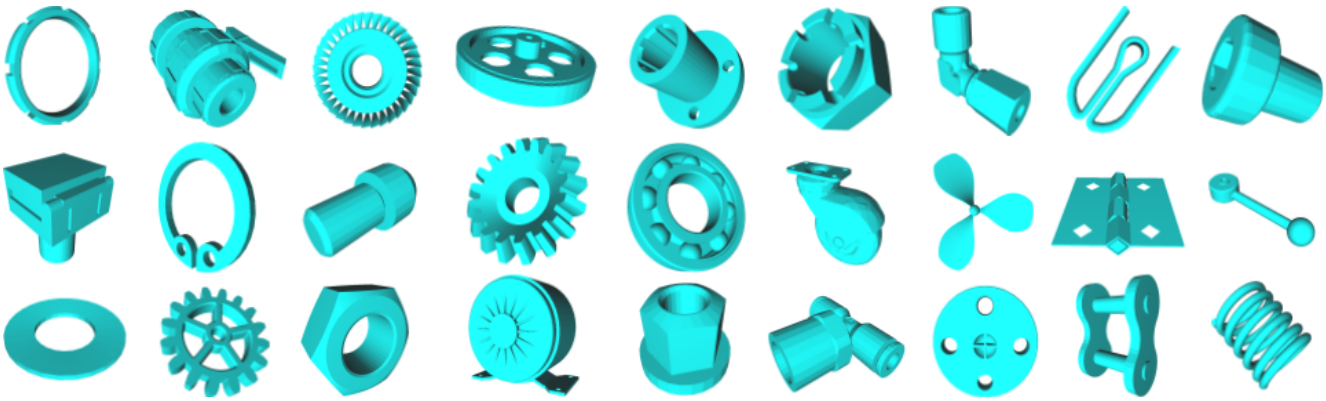

Figure 3: Randomly sampled mechanical components from the MCB.

Large-scale 3D object datasets Princeton Shape Benchmark (PSB) [38] is an early work that collected and annotated 3D objects for shape matching and classification benchmark. This work collected 3D polygonal models from the World Wide Web and also classified as to how the classes were constructed, 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 [29], shape retrieval [37], and shape reconstruction [7]. ModelNet [43] consists of two datasets, a 10-class dataset and a 40-class dataset, demonstrates a comprehensive clean collection of 3D CAD models of objects. PartNet [29] is a fine-grained, instance-level, and hierarchical parts dataset. It used 3D objects from ShapeNet and was annotated by 66 annotators.

Engineering shape datasets A Big Cad (ABC) Model Dataset [20] proposed one million Computer-Aided Design (CAD) models dataset without annotations. Engineering Shape Benchmark (ESB) [16] is an annotated engineering shape dataset. They 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 of 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) [35].

3. Properties of mechanical components

Mechanical components, shown in Figure 3, 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,

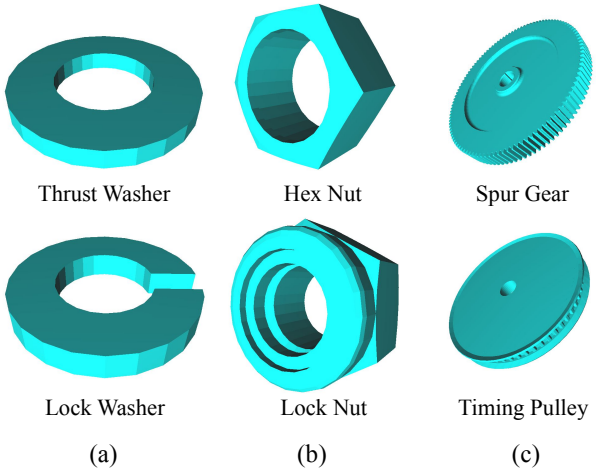

Figure 4: Examples of detail features make categorical changes.

resulting difficualty 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.

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 4 (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 4 (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 com-

Dataset	# Class	# Models	Type
ModelNet [43]	40	12,311	CO
ShapeNet [4]	3,135	+3,000,000	CO
ShapeNetCore	55	51,300	CO
ShapeNetSem	270	12,000	CO
PrincetonSB [38]	92	6,670	CO
PartNet [29]	24	26,671	CO
ABC [10]	N/A	+1,000,000	MC
AAD [2]	9	180	MC
ESB [16]	45	867	MC
MCB (Ours)	68	58,696	MC

Table 1: Comparison table of 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.

ing 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 4 (c). In contrast, with common objects like chairs, changing the shape of the legs or other detail features may not change the function of the chair. Because these characteristics do not appear in common object datasets [4, 43], the existing shape descriptors [33, 22, 37] 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*. [20] 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.

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 repository. To effectively annotate, CAD models are filtered and annotated using web-base 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 Warhouse[2], and GrabCAD[3]. 3D Warehouse and GrabCAD are large online open repository for professional designers, engineers, manufacturers, and students to share CAD models. They provide numerous CAD models with various classes, including mechanical components. The data from TraceParts are from industry defined components. By merging the models from different sources, we obtained more than 200,000 models before annotation.

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

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 2. 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 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 2. 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 that 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 [12]. We used projected images for annotating engineering parts.

Step 2: *Filtering.* We filter scrapped CAD models by deleting broken and duplicated models and capturing miss categorized models for re-annotation. Meta-information tagged in the process of scrapping step helps users to filter the data. File size can be used to remove the duplicated files, since duplicated files usually have the same size. 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.

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. 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.

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**

Table 2: The statistics of Mechanical Components Benchmark dataset.

5. Experiments

To analyze the depth of learning algorithms developed for common objects works on mechanical components, we benchmarked classification and retrieval task with three different representation: point could, projected views, and voxel grids. We use two NVIDIA GeForce RTX 2080Ti GPUs, i9-9900k CPU, and 64GB RAM for the experiments.

5.1. Statistic 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 2. To see the effect of data that has dense distribution, we build 2 forms of dataset for experiment:

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

Unlike other online CAD websites 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. The 3D models from TraceParts are aligned, but the others are not. 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.

5.2. Benchmark

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 [22], PointNet++ [33], and SpiderCNN [44]. For the multi-view base, we use MVCNN-DL [40] and RotationNet [18]. DLAN [10] and VRN [3] are used to evaluate voxel grids representation. For training each method, we use implementation codes and the hyper-parameters from the seven deep-learning algorithm papers and authors. For all the benchmark datasets, we randomly split the datasets into training set and test set as 80% and 20%, respectively. Training is conducted for each method to prevent initialization variation and report the best results.

Pcloud data Point cloud is the collections of points in Euclidean space. PointCNN [22] relaxes irregularity of

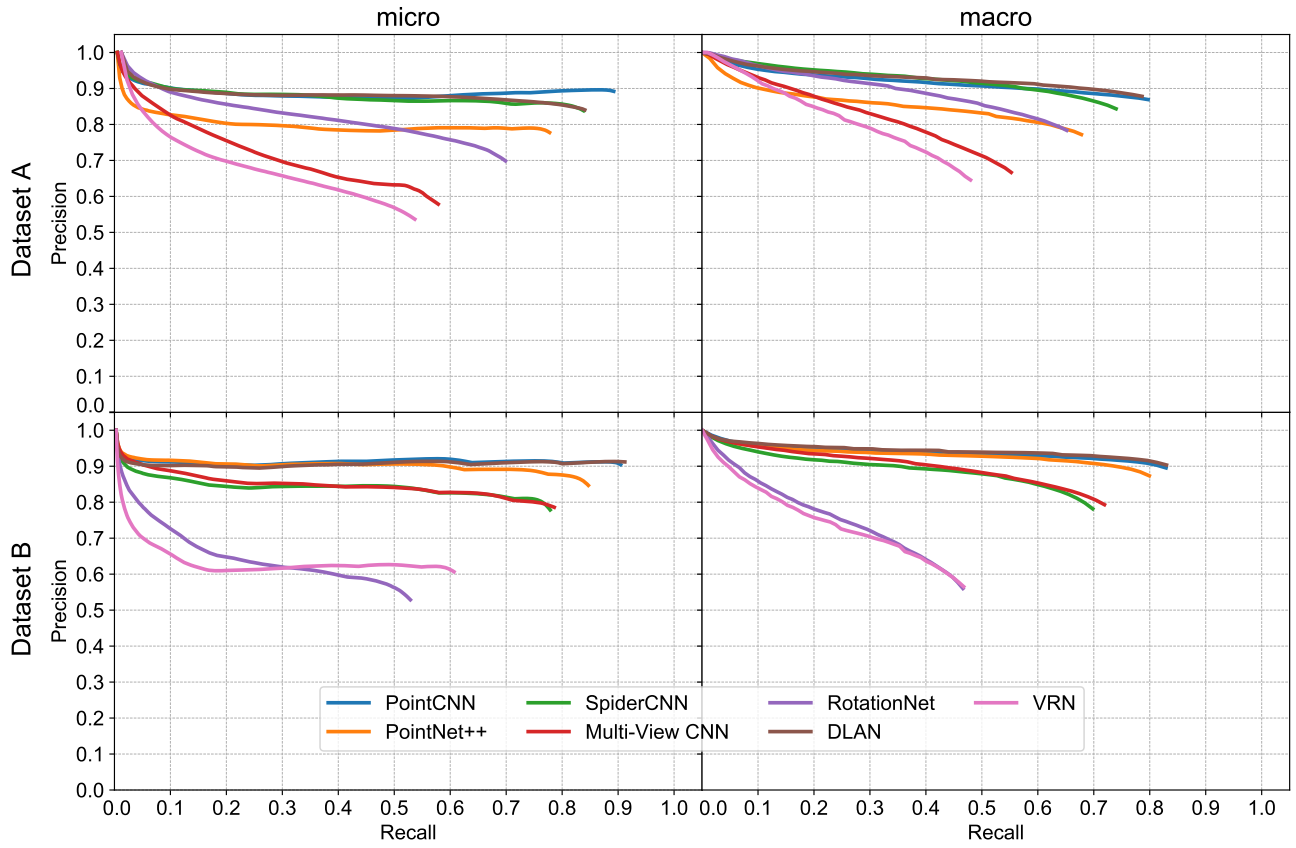

Figure 5: Precision-recall curve plots for retrieval with seven different methods. The PointCNN shows best retrieval results for dataset A and DLAN shows best retrieval results on dataset B.

Dataset	Method	micro P@N	micro R@N	micro F1@N	micro mAP	micro NDCG@N	macro P@N	macro R@N	macro F1@N	macro mAP	macro DCG@N
A	PointCNN [22]*	**0.892**	**0.892**	**0.690**	**0.889**	**0.898**	0.869	**0.797**	**0.833**	**0.886**	**0.854**
	PointNet++ [33]*	0.778	0.778	0.613	0.794	0.754	0.772	0.678	0.712	0.803	0.746
	SpiderCNN [44]*	0.839	0.839	0.669	0.867	0.793	0.844	0.741	0.776	0.877	0.812
	Multi-View CNN [40]⊠	0.579	0.579	0.488	0.657	0.487	0.667	0.552	0.585	0.735	0.641
	RotationNet [18]⊠	0.688	0.699	0.508	0.805	0.683	0.784	0.652	0.683	0.815	0.735
	DLAN [10]†	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
	Multi-View CNN⊠	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

Table 3: Summary table of evaluation metrics of shape retrieval benchmark for seven deep learning methods. Methods are grouped by their representation types. * indicates the method is based on point cloud representation and †indicates the method is based on volumetric representation. ⊠ indicates the method is based on images representation.

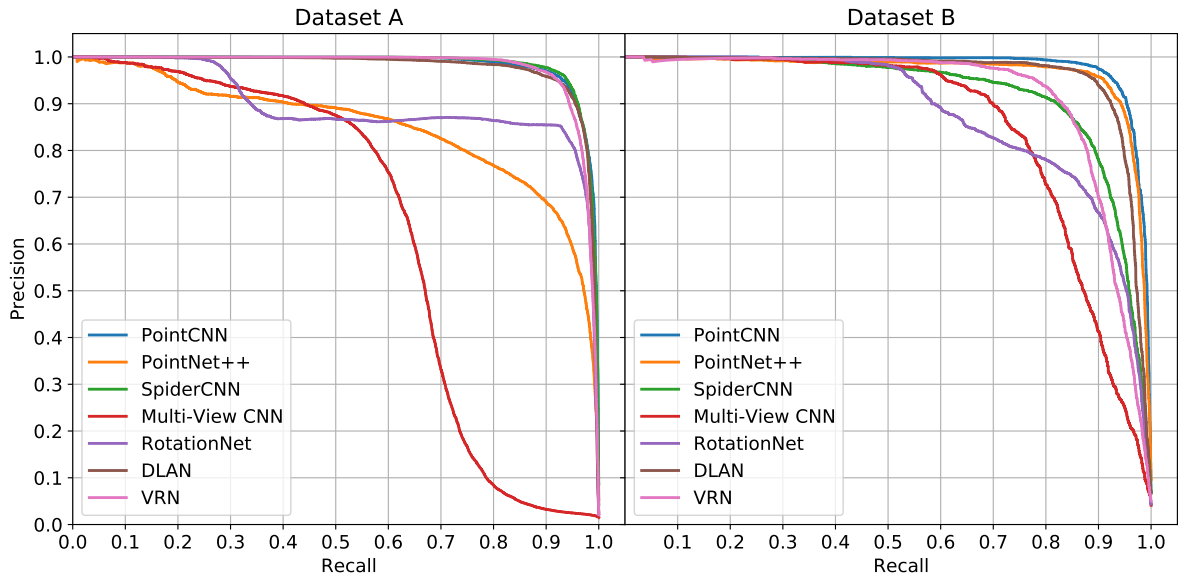

Figure 6: Precision and Recall (PR) curve of classification task. Left plot shows the PR curve of dataset A and right plot shows the PR curve dataset B. The RotationNet shows the best performance in terms of accuracy.

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

Table 4: 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.

point clouds by approximating the transformation matrix with multilayer perception, which simultaneously weights and permutes the input features for point cloud data feature learning. PointNet [32] 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++ [32] 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 [44] 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 [18, 39, 40] 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. MVCNN [39] shows a collection of multiple views of 3D objects is effective for learning their representations. MVCNN-ADL [40] further improves MVCNN with cross-modal distillation and adversarial inputs with a differentiable renderer.

Voxel Grids Three-dimensional objects can be discredited and represent in voxel grids, and voxel-based classifiers use voxel grids as their inputs. DLAN [10] 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 [25] learns the prob-

abilistic manifold of the underlying structure of voxelized 3D shapes with an auto-encoder in an unsupervised manner. VoxNet [27] 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.

5.2.1 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 [37]. Therefore, NDCG has a better ability to showing 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 retrieval result equally cross groups. Therefore, it eventually has the same P@N and R@N.

The summary results of all tested methods are given in Table 3. Corresponding precision-recall curves plots are given in Figure 5. We found that DLAN, voxel grids representation method, performed best among seven methods in both A and B datasets. However, VRN shows poor results in dataset B where the orientation is not aligned, even though it uses the same representation as DLAN. 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 the metrics have better performance in cross-category testing.

We observe that the performance of RotationNet and VRN drasmatically decrease on dataset B compared to dataset A. This is because the orientation of objects are aligned on dataset A but not on B. This is also observed in the common objects [37]. DLAN and PointCNN performs well in both datasets, 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 in both datasets.

5.2.2 Classification benchmark

We use four metrics, mean accuracy over objects, average accuracy per class, F1-score and Average Precision (AP) for the classification task and plotted precision-recall curves. We use macro method for F1 and AP. AP metrics are used to compare the network performance across dataset A and B. We used F1 score to measure the balance between precision and recall. The results are available in Table 4 and Figure 6. Overall, RotationNet outperforms the other methods in both dataset A and B. In point cloud methods, PointCNN shows the best performance for both datasets, and SpiderCNN much improved in the classification tasks after merging data from manufacturing companies. RotationNet performs well on both mean accuracy per class and per objects. However, PointCNN performs best for the AP. This is because PointCNN leverage spatially-local correlation and mechanical components are sensitive to the local changes.

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 application of deep learning algorithms 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. Accuracy for every class and network is specified on supplementary material. 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 these objects. From this motivation, we will work on developing a network that can extract both specificity and generality of shapes using our MCB dataset.

6. Conclusion

We propose a large-scale mechanical component benchmark with annotations. For the creation of the dataset, we develope an annotation framework that enhances the efficacy of the annotation and filtering processes. We perform shape classification and retrieval experiments with even deep-learning shape classification methods, which are designed to classify common objects. We find that view-based and voxel grids presentations methods perform poorly on unseen orientation shapes of mechanical components. However, DLAN, a voxel-based method, performs well on unseen orientation objects since it has orientation invariant property. The creation of MCB and experiment results can be used for the development of data-driven algorithms on mechanical components. For the future work, the development on dedicated data-driven feature descriptor for mechanical components will be conducted. Additionally, the MCB can be expanded to include other meta-data and this can become a knowledge network for supply chain and product life cycle.

References

[1] Open asset import library. http://www.assimp.org/. Accessed: 2019-11-10. 4

[2] Dmitriy Bespalov, Cheuk Yiu Ip, William C Regli, and Joshua Shaffer. Benchmarking cad search techniques. In *Proceedings of the 2005 ACM symposium on Solid and physical modeling*, pages 275–286. ACM, 2005. 3, 4

[3] Andrew Brock, Theodore Lim, James M Ritchie, and Nick Weston. Generative and discriminative voxel modeling with convolutional neural networks. *arXiv preprint arXiv:1608.04236*, 2016. 5, 6, 8

[4] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. ShapeNet: An Information-Rich 3D Model Repository. Technical Report arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015. 1, 2, 3, 4

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018. 1

[6] Chiho Choi, Sangpil Kim, and Karthik Ramani. Learning hand articulations by hallucinating heat distribution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3104–3113, 2017. 1

[7] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European conference on computer vision*, pages 628–644. Springer, 2016. 3

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1

[9] Kevin Ellis, Daniel Ritchie, Armando Solar-Lezama, and Josh Tenenbaum. Learning to infer graphics programs from hand-drawn images. In *Advances in neural information processing systems*, pages 6059–6068, 2018. 1

[10] Takahiko Furuya and Ryutarou Ohbuchi. 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*, pages 171–178. ACM, 2015. 4, 5, 6, 7

[11] Takahiko Furuya and Ryutarou Ohbuchi. Deep aggregation of local 3d geometric features for 3d model retrieval. In *BMVC*, pages 121–1, 2016. 1

[12] Christophe Geuzaine and Jean-François Remacle. 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. 4

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 1

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[15] Jida Huang, Tsz-Ho Kwok, and Chi Zhou. Parametric design for human body modeling by wireframe-assisted deep learning. *Computer-Aided Design*, 108:19–29, 2019. 1

[16] Subramaniam Jayanti, Yagnanarayanan Kalyanaraman, Natraj Iyer, and Karthik Ramani. Developing an engineering shape benchmark for cad models. *Computer-Aided Design*, 38(9):939–953, 2006. 3, 4

[17] Xu Jia, Efstratios Gavves, Basura Fernando, and Tinne Tuytelaars. Guiding the long-short term memory model for image caption generation. In *The IEEE International Conference on Computer Vision (ICCV)*, December 2015. 1

[18] Asako Kanezaki, Yasuyuki Matsushita, and Yoshifumi Nishida. Rotationnet: Joint object categorization and pose estimation using multiviews from unsupervised viewpoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5010–5019, 2018. 1, 5, 6, 7

[19] Sangpil Kim, Nick Winovich, Hyung-Gun Chi, Guang Lin, and Karthik Ramani. Latent transformations neural network for object view synthesis. *The Visual Computer*, pages 1–15, 2019. 1

[20] Sebastian Koch, Albert Matveev, Zhongshi Jiang, Francis Williams, Alexey Artemov, Evgeny Burnaev, Marc Alexa, Denis Zorin, and Daniele Panozzo. Abc: A big cad model dataset for geometric deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9601–9611, 2019. 3, 4

[21] Tejas D Kulkarni, William F Whitney, Pushmeet Kohli, and Josh Tenenbaum. Deep convolutional inverse graphics network. In *Advances in neural information processing systems*, pages 2539–2547, 2015. 1

[22] Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Xinhan Di, and Baoquan Chen. Pointcnn: Convolution on x-transformed points. In *Advances in Neural Information Processing Systems*, pages 820–830, 2018. 4, 5, 6

[23] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[25] Shikun Liu, Lee Giles, and Alexander Ororbia. Learning a hierarchical latent-variable model of 3d shapes. In *2018 International Conference on 3D Vision (3DV)*, pages 542–551. IEEE, 2018. 7

[26] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[27] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition.

In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928. IEEE, 2015. 8

[28] George A Miller. *WordNet: An electronic lexical database*. MIT press, 1998. 1

[29] Kaichun Mo, Shilin Zhu, Angel X Chang, Li Yi, Subarna Tripathi, Leonidas J Guibas, and Hao Su. Partnet: 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*, pages 909–918, 2019. 1, 2, 3, 4

[30] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodola, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model cnns. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1

[31] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv preprint arXiv:1606.02147*, 2016. 1

[32] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017. 7

[33] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 1, 4, 5, 6

[34] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016. 1

[35] William C Regli, Cheryl Foster, Erik Hayes, Cheuk Yiu Ip, David McWherter, Mitchell Peabody, Yuriy Shapirsteyn, and Vera Zaychik. National design repository project: a status report. In *International Joint Conferences on Artificial Intelligence (IJCAI), Seattle, WA, Aug*, pages 4–10, 2001. 3

[36] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 1

[37] Manolis Savva, Fisher Yu, Hao Su, Asako Kanezaki, Takahiko Furuya, Ryutarou Ohbuchi, Zhichao Zhou, Rui Yu, Song Bai, Xiang Bai, et al. Large-scale 3d shape retrieval from shapenet core55: Shrec'17 track. In *Proceedings of the Workshop on 3D Object Retrieval*, pages 39–50. Eurographics Association, 2017. 2, 3, 4, 8

[38] Philip Shilane, Patrick Min, Michael Kazhdan, and Thomas Funkhouser. The princeton shape benchmark. In *Proceedings Shape Modeling Applications, 2004.*, pages 167–178. IEEE, 2004. 1, 2, 3, 4

[39] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *Proceedings of the IEEE international conference on computer vision*, pages 945–953, 2015. 1, 7

[40] Jong-Chyi Su, Matheus Gadelha, Rui Wang, and Subhransu Maji. A deeper look at 3d shape classifiers. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 5, 6, 7

[41] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017. 1

[42] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015. 1

[43] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920, 2015. 1, 2, 3, 4

[44] Yifan Xu, Tianqi Fan, Mingye Xu, Long Zeng, and Yu Qiao. Spidercnn: Deep learning on point sets with parameterized convolutional filters. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 87–102, 2018. 5, 6, 7