



DTV-CNN: Neural network based on depth and thickness views for efficient 3D shape classification

Qingfeng Xia

Culham Centre for Fusion Energy, United Kingdom Atomic Energy Authority, OX14 3DB, United Kingdom

ARTICLE INFO

Keywords:

3D shape classification
Depth map
Neural network
CAD
Intelligent engineering design
AI
Thickness
CNN
EDA

ABSTRACT

Fast and effective algorithms for deep learning on 3D shapes are keys to innovate mechanical and electronic engineering design workflow. In this paper, an efficient 3D shape to 2D images projection algorithm and a shallow 2.5D convolutional neural network architecture is proposed. A smaller convolutional neural network (CNN) model is achieved by information enrichment at the preprocessing stage, i.e. 3D geometry is compressed into 2D “thickness view” and “depth view”. Fusing the depth view and thickness view (DTV) from the same projection view into a dual-channel grayscale image, can improve information locality for geometry and topology feature extraction. This approach bridges the gap between mature image deep learning technologies to the applications of 3D shape. Enhanced by several essential scalar geometry properties and only 3 projection views, a mixed CNN and multiple linear parameter (MLP) neural network model achieves a validation accuracy of 92 % for ModelNet10 mesh-based dataset, while the training time is one order of magnitude less than the original multi-view CNN approach. This study also creates new 3D shape datasets from 2 open source CAD projects. Higher validation accuracy is obtained for realistic CAD datasets, i.e. 97 % for FreeCAD’s mechanical part library and 95 % for KiCAD electronic part library. The training cost reduces to tens of minutes on a laptop CPU, given the smaller input data size and shallow neural network design. It is expected that this approach can be adapted for other machine learning scenarios involved in CAD geometry.

1. Introduction

1.1. Machine learning for CAD applications

The success of convolution neural network in 2D image classification and recognition encourages active investigation of machine learning on 3D objects. Machine learning of 3D geometry classification and recognition, is one of the enablers for digital engineering, in the author’s words, the automated and intelligent engineering design workflow. The design of a complex machine/product needs several design iterations. Without a continuous integration workflow as in software engineering, the overall design period of an aero-engine model or a fusion reactor demonstration facility, would be much longer than a large-scale software project. Artificial intelligence in the context of digital engineering can revolutionize the existing design-simulation workflow.

There are various 3D data representation formats, point set, surface mesh, text description as in CAD software, etc. 3D geometry dataset from CAD software is different from mesh and point set, with unique features such as high precision, complicate topology. First of all, 3D geometry is bounded by surface precisely; and surface area and volume can be calculated to a high precision, in contrast to

E-mail address: xiaqingfeng1@huawei.com.

<https://doi.org/10.1016/j.heliyon.2023.e21515>

Received 29 January 2023; Received in revised form 20 September 2023; Accepted 23 October 2023

Available online 31 October 2023

2405-8440/© 2023 Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

triangle mesh approximation. Mesh can be generated from CAD geometry with information loss, while the geometry reconstruction from mesh is possible but time-consuming and imperfect. Secondly, there are meta data embedded with geometry. For example, the industrial standard file exchange format STEP v214, has physical material property like density embedded. Recent version of STEP v242 has enriched meta data such as tolerance and assembly constraint. Last but not the least, there are large number of standardized parts such as bolts and nuts. Machine design is a mixture of standard parts and in-house designed components.

Deep learning of CAD generated geometry files for part and assembly is less reported than surface mesh-based 3D object datasets, due to the lack of publicly accessible dataset. CAD data are exchanged by STEP file format, a complex text file format using the Express language to describe geometry and topology. It is more like a data markup language, and thus not suitable to apply existing language deep learning model. Existing studies usually convert STEP geometry into other type of 3D representation. For example, MVCNN++, the enhanced Multi-View Convolutional Neural Networks, improved accuracy by 5.9 % for CAD shapes [1].

High accuracy 3D geometry classification and retrieval is a key enabler for various innovations in computer-aided design and simulation workflow, eventually leading to intelligent engineering design. For example, generative neural networks can generate a prototype based on existing CAD models of same kind, as the starting point of design-simulation iteration [2]. 3D shape without meta data could be searched and identified in product database, just like image search [3]. Simulation-related setup such as boundary condition for design verification can be recommended automatically, based on part functional classification and rules learned from previous simulation datasets. Performance evaluation at the post-processing stage of CAE software, can also be automated. Recent work on geometry feature recognition has extended into the domain of computer-aided manufacturing (CAM) application, with in-house generated CAD dataset, see Refs. [4,5].

Aiming at the continuous integration (CI) pipeline workflow like software engineering, product design should be conducted smoothly and intelligently for material selection and topology optimization. Eventually, an automated and intelligent engineering design-simulation workflow will be enabled to reduce or even eliminate engineers' intervention in the engineering design process, which will significantly reduce the time to market for products and mark a new era for computer-aided engineering design.

1.2. Challenges of deep learning on 3D shapes

In addition to meta-data rich 3D geometry datasets, efficient deep learning models are desired to accelerate the training on 3D datasets. Deep learning is a time-consuming task, especially for 3D geometry which has one more dimension than 2D images. Although deep learning on 3D dataset has become practical using the powerful hardware, reducing the computation cost is still a priority to drive the wider adoption in desktop and mobile platforms.

1.3. Organization of this paper

This paper focuses on 3D shape classification at a single part or component (clustering of several parts to form a functional entity) level, but capable of classifying product level with hundreds or more parts. For the latter cases, high-resolution 2D images could be a more efficient approach for classification.

Aiming to reduce the computation cost for 3D part classification, optimization has been carried at both preprocessing and training stages. At the preprocessing stage, an innovative 2.5D projection view, i.e. thickness view, is proposed and its generation algorithm is illustrated. The input data byte size for each geometry is only 2 % of 12 view MVCCN model. According to Shannon information entropy theory, information compression at the input stage can significantly reduce the neural network trainable parameters. In the training stage, the fusion of depth and thickness maps as a single 2D two-channel image is adopted as model input, and the efficiency of this proposed model is benchmarked with existing models on ModelNet datasets.

Another major contribution of this paper is the exploration and preparation of publicly available CAD datasets for deep learning related to mechanical and electronic design application. The performance of this proposed deep learning model is studied against these newly created CAD datasets.

2. Literature review

2.1. Overview of 3D shape classification

Before the success of deep learning on 2D image classification and recognition, machine learning on 3D shape was even less reported [6]. Recent progress in convolution neural network and the autonomous driving applications have promoted the research in machine learning on 3D shapes. Existing machine learning approaches can be grouped by how 3D data are represented; most of deep learning models take only one of input data types below:

- Multi-view RGB colorful images: a sequence of images from different view angles, e.g. PANORAMA-NN.
- RGB-D: 2D color picture with an extra channel for depth information (how far it is the pixel to the viewer).
- Projection map images: various mapping from 3D shape into 2D image format.
- Volumetric: e. g. voxelization [7] as dense or sparse 3D array.
- Point cloud: array of point coordinate, typically obtained from light detection and ranging (LiDAR) widely equipped in autonomous driving vehicle [8].
- Polygonal mesh: surface mesh widely used in 3D graphics rendering, such as ModelNet10 and ModelNet40 [9].

- Primitive-based descriptor: scalar geometrical properties such as volume, bounding box, etc.

Combination of different input data formats is not widely explored, due to the cost of preprocessing and training time. However, there are some examples like combining point cloud and multiple view image data [10].

The Princeton ModelNet project has been maintaining ModelNet Benchmark Leaderboard for 3D shape classification based on ModelNet dataset at: <https://modelnet.cs.princeton.edu/>

2.2. Multiple viewpoint images

A series of 2D images from different viewpoints are widely used, since this simulates human eyes' observation of a 3D shape from distinct perspective. The rendered image highly depends on lighting configuration, since geometry details are distinguished by color and brightness change. The very first multiple-view convolution neural network (MVCNN) model [11] uses 12 views for non-oriented shape, and further 80 views to improve accuracy. The large input byte size per sample slows down the training. A comparison of PVRnet [12] with MVCNN, GV-CNN, has showed observable improvement from 4 views to 8 views, but increasing the view count to 12 and 16 views does not help [13].

Three-dimensional shape recognition with a multi-view approach has been reviewed for research before year 2021 [14], most multiview-based methods ignore the correlations of multiple views or suffer from high computational cost. After the proposed MVCNN model [11], there are several enhanced models, such as long short-term memory (LSTM) to exploit the correlative information from multiple views [15], MVCNN-MultiRes [16] using 20 views of 3 kinds of image resolution can improve accuracy from 92 % to 94 % for the ModelNet40 dataset. Instead of maximum view pooling in MVCNN, bilinear pooling and a harmonizing layer [17] improve the accuracy to nearly 95 % for both ModelNet10 and ModelNet40 datasets. Latent-MVCNN [18] has shown predefined views' advantage over random-selection, and utilizes latent vector to improve view relevance, to gain a higher accuracy.

2.3. Projection methods

In conventional mechanical engineering drawing, three 2D views are used to represent a 3D object without any loss of information. In order to reduce the input data size, i.e. view count, for MVCNN, different 2D representations of 3D objects have been investigated [19]. Some methods result in human readable projection image such as depth-map; some other are not visually comprehensible, leading to difficulty to visually evaluate the quality of mapping.

SPNet [12] uses stereographic projection and comparatively studies with other 3D to 2D project methods. Single view can achieve about 90 % accuracy. Some other projection method such as depth map [20] can improve the accuracy using multiple views; 6 depth maps with a resolution of 320×240 attains an accuracy of 87.8 % on ModelNet40 dataset. In addition to depth map and volumetric (voxel density) map (projecting voxel to 2D image by counting voxel), and surface local principal curvatures has been captured as the third data source [21].

2.4. 3D voxel volumetric data format

The 3D voxel data format is widely used in 3D printing and volumetric visualization. This is an intuitive way to represent 3D object, but the memory usage is high. Voxnet [7] is the original voxel-based deep learning model, achieving an accuracy of 92 % on ModelNet10 dataset. In addition to equally-spaced voxelization, section views and slicing (DICOM) data format is also popular in medical imaging. Voxel-based data format has been applied in analysis and search of mechanical CAD-models [22]. Simplified voxel representation can be a good approach to understanding the topology, and further reduces the model parameter size. It is reported [23] a local phase 3D CNN uses only 11 % of the parameters to achieve the state-of-the-art accuracy on ModelNet dataset; the secret to such an accomplishment is feature extraction from 3D local blocks (e.g., $3 \times 3 \times 3$) and the modified 3D convolution.

The disadvantage of voxel representation lies in the storage and computational cost. Converting shape into dense 3D array, i.e. voxelization, consumes large amount memory and computational resource at the preprocessing stage. Nevertheless, 3D convolution, one dimension higher than convolution on 2D images, is more computation-demanding.

2.5. Point cloud

Computational cost on point cloud format is higher than 2D images. Point cloud data can also have multiple views from different projection axes to represent a 3D shape [13]. To classify shape with the point-cloud data format, point-to-vector capsule (PVC) network forms a 3D shape descriptor using 2 feature extraction steps followed by a dynamic routing algorithm [24]. Another work, dual-graph attention convolution network (DGACN) [25], fuses low-level extrinsic and high-level intrinsic graph features for point cloud classification. A major shortcoming for point cloud is the incapability to represent topology which is crucial for CAD geometry.

2.6. Surface mesh

3D geometry in surface mesh format, is vertex/point further equipped with connectivity information to form facets. Surface mesh is widely used in game engineering to fully define a 3D shape. ShapeNet dataset is prepared in surface mesh file format, and geometry for this popular dataset is further processed into other 3D reorientations mentioned above. However, direct usage of mesh triangulation as

input has been reported as MeshNet [26], which uses not only point coordinates but also triangle normal and interconnection information. Alternatively, MeshCNN [27] has designed a specialized convolutional operator for edge of 3D triangular mesh and a pooling operator further merging triangles while retaining the mesh topology; this MeshCNN is capable of mesh simplification, segmentation, etc.

2.7. Primitive-based: geometrical properties

3D geometry classification based on descriptors was popular before the great success of CNN in 2D image deep learning. Geometrical properties such as bounding box sizes, surface area and volume, are computationally cheap. For example, in ModelNet datasets, airplane and cup are two distinguish classes that can be distinguished by volume value, but volume and area properties do not help to understand the topology. There are some more complicate properties [28] like 3D spherical harmonics, shape distribution, refer to the "Survey on 3D shape descriptors" [29]. Light field distribution (LDF) is found efficient in part classification for a real mechanical assembly model. However, hundreds of geometrical parameters are needed to achieve a satisfied accuracy, which requires lots of computation resource in the preprocessing stage.

2.8. Graph convolutional network (GCN)

Graph convolutional network [30] has been widely used for social network analysis, and been extended to 3D shape analysis with mesh datasets, such as FeaStNet [31]. PointView-GCN [32] with multi-level GCNs can hierarchically collect the shape features of single-view point clouds and their multi-view relations, achieving a high accuracy of 95 %, i.e. 5 % higher than single view point cloud approach. The view-GCN [33] understands 3D shape by graph representation of multiple views, with each view as a graph node undergoing local convolution and coarsening. Recently, those authors have improved the model's robustness regarding shape spatial transformation, with view-GCN++ [34] with local attentional graph convolution operation and rotation robust view-sampling operation.

2.9. Combination of submodels

Each approach listed above has its own advantage in shape analysis, and it is natural to combine them for better accuracy and robustness. PVNet [35] uses a joint CNN of point cloud and multiple view data input for 3D shape analysis; an embedding attention fusion scheme is introduced to correlate information from different views. The network architecture of NLGAT [36] is a combination of a global relationship network and a structural network for point cloud data by a fully connected layer. Nevertheless, primitive-based descriptors can be mixed, since they are cheap to calculate during preprocessing and do not significantly increase the total trainable parameter count. Bag of feature [37] uses a method other than deep learning for 3D shape classification.

3. 3D shape datasets

Large 3D datasets are crucial for deep learning research on 3D object classification and/or retrieval. Driven by the fast development of autonomous driving and robotics, real-world shape datasets obtained from scanning hardware are abundant, such as ScanObjectNN [38] with 2902 3D objects in 15 categories. On the other hand, CAD datasets are rare in the public domain for deep learning purpose. In this study, new 3D geometry datasets in native CAD file formats are explored and created from open-source CAD projects.

3.1. Mesh-based datasets: ModelNet and ShapeNet

ModelNet10 [39] dataset comes with mesh format, not CAD geometry file format like STEP. ModelNet10 includes ten categories of 3991 and 908 models into training and testing partitions respectively. ModelNet40 has 40 categories with a total sample of 12 thousand. There are other 3D surface mesh based datasets such as Thing10K [40] for 3D printing models, but ModelNet datasets are de facto for shape classification applications with abundant benchmarking result. ShapeNet [9] is another large-scale 3D shape dataset with both point cloud and surface mesh formats; PartNet [41], ShapeNetCore and ShapeNetSem are subsets of ShapeNet. Among them, ShapeNetCore dataset contains 51,300 samples in 55 categories.

3.2. Reported 3D CAD datasets

There are several online CAD part hosting websites like Grabcad and TraceParts. A dataset with 2354 elements among 15 sub-categories has been used in part classification using feature parameters [29] is collected from those two sources. However, no download link is provided for this 3D geometry dataset, except for webpage summary and statement for copyright restriction. Drexel dataset is another CAD dataset, but without valid download link neither. ABC-dataset [42] has a million CAD samples without labels, which is not suitable for this part classification study.

Recent research on CNN for view-based 3D object retrieval [43] uses 3 datasets. ETH 3D object dataset has 80 objects belonging to 8 categories. The second dataset of NTU60 3D model [44] has 549 objects belonging to 47 categories. The last dataset, MVRED 3D category dataset [45], has 505 objects in 61 categories. An average of about 10 samples per category suggests that those datasets is not ideal for shape classification. The most recent 3D shape dataset is OmniObject3D [46] with 6 thousand scanned objects in 190

categories, sharing common categories with popular 2D datasets like ImageNet. The paper for OmniObject3D also includes a statistic comparison with other 10 3D shape datasets. Due to the lack of a suitable CAD dataset, some research, like 3D feature recognition, generates its own CAD dataset, seen in Refs. [4,5].

3.3. Mechanical CAD part library from FreeCAD

The standard part library for FreeCAD, an influential open source mechanical CAD project <https://github.com/FreeCAD/FreeCAD-library>, has been organized as a new dataset in this study. This library features a well-structured folder hierarchy, and labels can be retrieved from subfolder names. The labeling matches the well-established part classification based on functionality, as in machine design text book [47]. However, some categories do not have enough samples suitable for machine learning purpose. Given the threshold of minimum sample count as 20 for each group, 10 groups of 1838 mechanical parts can be collected from FreeCAD library dataset, see Table 1.

3.4. Electronic 3D part library from KiCAD

KiCAD is an open source schematic capture and PCB design software; its PCB layout features a well-organized collection of 3D shapes for common electronic categories, from resistors, capacitors to various connectors and packages, see the whole dataset at: <https://kicad.github.io/packages3d/>. Using the same threshold of a minimum sample count of 20, 2029 samples in 18 categories have been collected, see Table 2. Some electronic components have two mounting options, surface mount design (SMD) and through-hole technology (THT); it was decided to merge them into single category. Both package and connector categories have a few subtypes, only some common categories were selected due to data availability.

Both FreeCAD and KiCAD part libraries are distributed with a permissive license of Creative Commons – Attribution, which makes both datasets publicly accessible for deep learning in mechanical and electronic design. In addition to their native file formats, the industrial standard CAD file format like STEP, is provided for data exchange. The original FreeCAD and KiCAD datasets are well-organized in file system with subfolder name as the category. Especially, KiCAD's part library covers most common discrete electronic components.

To dispense the labor for preprocessing, the preprocessed image data in NumPy file and metadata in JSON file format, together with all preprocessing scripts scheduling view generation and image collection, for FreeCAD and KiCAD datasets can be downloaded at <https://github.com/AICAE/3DPartClassification/tree/master/data>.

Admittedly, FreeCAD and KiCAD datasets have only 1/3 to 1/2 sample count as the ModelNet10 dataset, while they are more accessible than other CAD datasets for 3D geometry classification applications. Compared with commercial CAD software, neither FreeCAD nor KiCAD part library is full-fledged. However, the library scale is ever-growing with an active user community, and they will remain as promising 3D datasets freely available for deep learning applicable in mechanical and electronic design.

4. Preprocessing

Reducing computation cost has been the primary goal in this study. Given the fact that 3D CAD geometry is not fed directly into neural network, the smaller neural network input data size means less model parameters to train. The first priority is given to the preprocessing stage, compressing 3D geometry into high entropy information, i.e. fusion of depth and thickness, inspired by Shannon's information theory.

4.1. Preprocessing algorithm

4.1.1. Alignment and bounding box selection

Research by Ref. [48] claimed that shape orientation plays an important role in 3D recognition for voxel 3D data format. Meanwhile, orientation is also crucial for multiple view based CNN models, since "pose normalization" [3] for each 3D shape is essential to generate comparable views independent geometrical transformations such as rotation. CAD parts are usually axis-aligned with the

Table 1
Summary of FreeCAD library 3D part dataset.

id	category name	samples
0	Cylinders	406
1	Bolts & Screws	431
2	DIN1025-Profiles	90
3	EN10056 Angle Bars	155
4	EN10058 Flat steel bars	214
5	EN10059 Square steel bars	35
6	EN10060 Round steel bars	67
7	EN10219 Rectangular Hollow Sections	163
8	EN10219 Square Hollow Sections	145
9	Washers	132

Table 2
Summary of KiCAD 3D part library selected for this study.

id	category name	samples
1	Button Switch (SMD & THT)	163
2	Capacitor (SMD & THT)	460
3	Connector Molex	89
4	Connector Phoenix MC	180
5	Connector Pin Header	156
6	Connector Pin Socket	246
7	Crystal	99
8	Diode THT	89
9	Inductor THT	139
10	LED THT	75
11	Package BGA (ball-grid array)	78
12	Package DIP (dual-in-line package)	202
13	Package SOP (small outline package)	184
14	Relay THT	47
15	Resistor THT	105
16	Terminal Block Phoenix	86
17	Transformer THT	29
18	Varistor	100

design space coordinate system, while different designers may have different preference for a principal axis. A more specific description for shape orientation can be given as, all the airplane shape should have head forwarding along the same axis with wing span aligned with the identical second axis. ModelNet10 is an oriented dataset, in contrast to ModelNet40. Orion offers an aligned ModelNet40 dataset at <https://github.com/lmb-freiburg/orion>, as well as the algorithm of model alignment.

For an unaligned dataset, re-orientation is an important preprocessing task; hints of shape transformation can be gained from oriented bounding box and principal rotation axis/momentum. Oriented bounding box (OBB) may be not aligned to axes in the design space, but gives the minimized bounding box volume. OBB may be referred as minimal bounding box (MBB) in some other literature. Principal component analysis is a popular algorithm for shape alignment based on OBB. In addition, a multi-view transformation network (MVTN) [49] has been reported to be able to improve the rotation robustness of any multiple view based neural network models on 3D shape classification on ModelNet40 dataset.

4.1.2. Depth and thickness view (DTV) in a single image

Traditionally, mechanical engineers understand a 3D geometry from a series of 2D projection views, much fewer than 12 or 76 views in MVCNN. The typical combination of 3 orthogonal views are front, top and left views, see example drawings in Fig. 1. With the carefully-selected orientation and orthogonal projection, engineers can reconstruct the 3D shape in mind. However, mechanical drawing is always a black-and-white binary image, bearing only information for curves. If more information can be embedded into a gray-scale image, CNN will be benefitted for feature enrichment.

Depth map has been widely used in 3D shape classification. Meanwhile, thickness view bears more topology information, not only edge and corner, but also internal feature. In this study, both forward and backward depth and thickness maps are generated using the

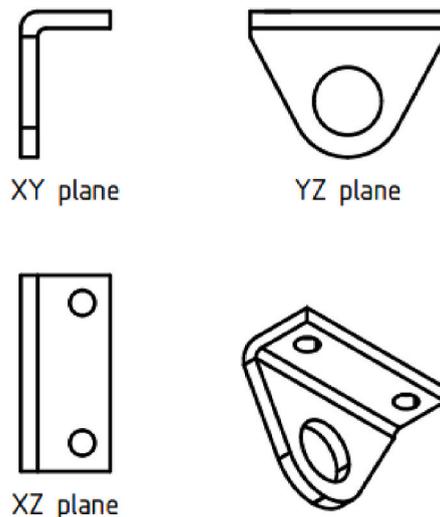


Fig. 1. Mechanical drawing with 3 orthogonal axis-aligned views and one tri-axis view (bottom right).

same projection view and mapped to the same image coordinates and saved into a single image file. In other words, depth and thickness is treated as different color channels of one RGB image.

The key innovation of this paper, is the fusion of thickness and depth maps originated from the same projection view. This improved information locality enables more effective feature extraction by CNN. For a part without any internal void/cavity, a single depth-thickness-view (DTV) has the sufficient information to reconstruct the 3D part; while 3 views can increase the efficacy of topology learning by CNN and deal with more complex geometry. The view count of 3 is widely used in mechanical drawing, and therefore, selected as the baseline in this research, while more views can be fed into CNN, e.g. 3 orthogonal views after rotating the shape 45° around Z-axis. Fig. 2 demonstrates three orthogonal projection directions for a mechanical part. Each row presents a group of images from the same projection direction.

We can see these images bear more information than the pure curves outlining only the boundaries of part without knowledge of interior region. The first two columns are single channel grayscale image for depth map and thickness views respectively, painted with pseudo color scale. The rightmost column in Fig. 2 is the fusion of both depth map (the forward) and thickness map into a dual-channel image, and painted as red and green channel of a RGB image. The third/blue channel is kept as always zero, since void space is represented by dark pixel. CNN accepts both single-channel grayscale and multi-channel images; the latter has better correlation between depth and thickness information.

4.1.3. DTV image generation algorithm

In this study all the shapes have been oriented (aligned), and axis-aligned bounding box (AABB) is used to select the region of interest. Through some preliminary study, it was found that a cube with the length as the longest edge of axis-aligned bounding box can generate better accuracy, because it maintains the aspect ratios for any given projection view. However, resolution loses if aspect ratios deviate from unity, for example, there will be a large dark area on image for a slim part.

In contrast to the time-consuming voxelization process of volumetric information retrieving [21], thickness and depth image can be directly calculated from shape and ray intersection. Thickness projection image can be achieved by geometrical Boolean operation; the thickness at any given ray path can be summed from all the intersected line segments. The CAD geometry, and the equally-spaced ray grid mapped to pixels in thickness image, are illustrated as the left section of Fig. 3. The intersected line segments between the projection ray grid and the target geometry are given in the right side of Fig. 3, using the same viewport.

Boolean intersection is a time-consuming operation, therefore, a highly efficient thickness view projection algorithm based on tessellation has been designed, which is nearly two orders of magnitude faster.

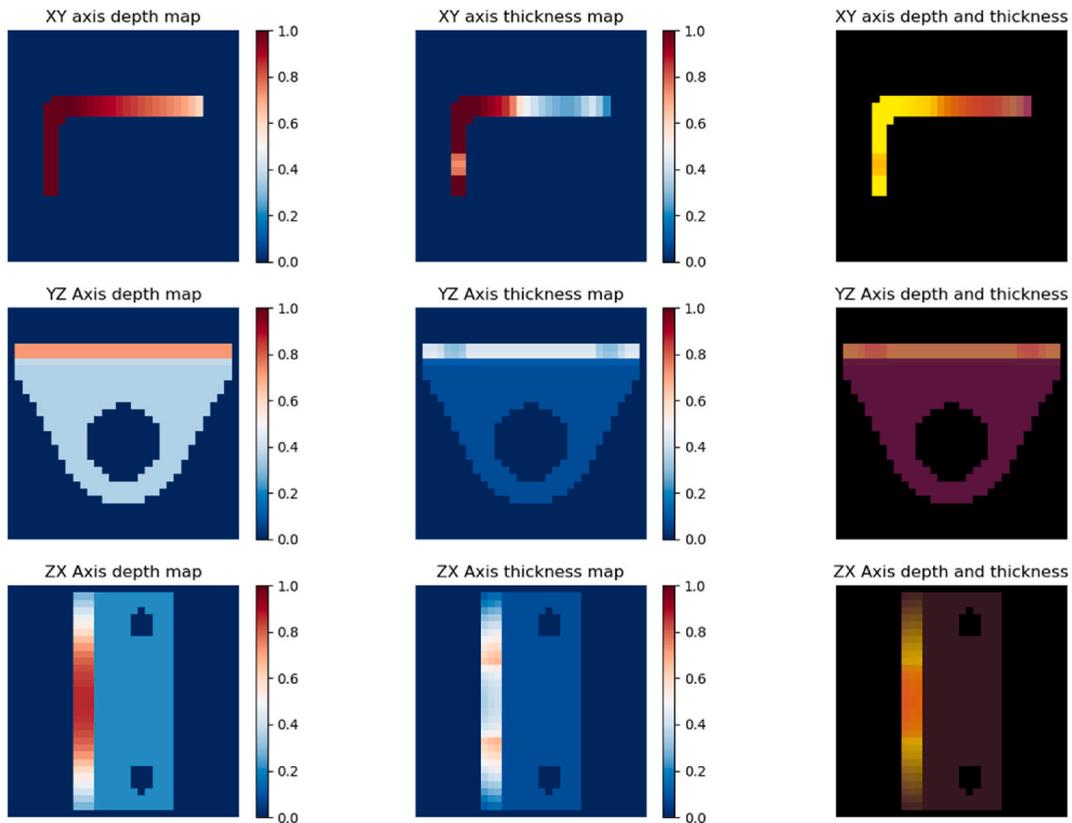


Fig. 2. Sample of depth view and thickness views of the part from three orthogonal projections.

4.1.4. Thickness projection by ray-mesh intersection

Boolean intersection operation is robust but time-consuming to calculate the common parts of the geometry and its ray grid. Therefore, a more efficient way to calculate the thickness is proposed in this paper, analogous to the well-established ray-intersection algorithm. Furthermore, this algorithm is suitable for GPU acceleration.

Three orthogonal project views aligned with its oriented coordinate system are selected to generate a ray-grid. The collection of rays is organized into an equally-spaced 2D grid, while the 3D shape is tessellated into surface triangle mesh. The C++ source code is available inside the repository: <https://github.com/AICAE/3DPartClassification>, residing in occProjector folder, while the pseudo code for thickness and depth view calculation for a given projection direction is given,

Algorithm based on tessellation and ray grid intersection

```

1:                                     Input: mesh; ray grid for a given projection.
2:                                     Output: thickness and depth view as matrix  $T$  and  $D$ 
3: initialize a matrix of variable-length vector  $M$  element-wise with an empty vector  $M_{ij} = []$ 
4: foreach triangle (v0, v1, v2) in mesh do
5:   find ray index range [sitar, iend] and [jstart, jend], this triangle may intersect according to coordinate;
6:   foreach  $i$  in [istart, iend] do
7:     foreach  $j$  in [jstart, jend] do
8:        $h$ , intersected = intersect(triangle, ray[i,j])
9:       if intersected == true do
10:        push  $h$  into the scalar vector at  $M_{ij}$ 
11:       end if
12:     end foreach index  $j$  in range
13:   end foreach index  $i$  in range
14: end foreach triangle in mesh
15: sort ascendingly for each vector [ $h_0, h_1, \dots$ ] in  $M$ 
16:  $h = \sum_k^{1,3,\dots,n-1} (h_k - h_{k-1})$ 
17: where  $k$  is the zero-based vector index
18:  $T_{ij} = \frac{h}{h_{\max} - h_{\min}}$  scale  $h$  into the range of [0, 1],
19: where  $h_{\max}$  and  $h_{\min}$  are ray coordinate limits;
20:  $d = \max([h_1, h_2, \dots])$  or  $\min([h_1, h_2, \dots])$ ,
21: depends on axis direction for depth view
22:  $D_{ij} = \frac{d - h_{\min}}{h_{\max} - h_{\min}}$  scale depth into the range of [0, 1]

```

In case of XY view, only the Z-axis coordinate of the intersection points, corresponding to h in the code above, needs to be appended into the vector M_{ij} . Each triangle in the meshed shape can only intersect with a few rays, whose row-column index range could be calculated by triangle vertex coordinate. For manifold 3D shape, it is evident that the scalar count in each vector should be an even number. For the situation of zero intersection, the thickness and depth value are set as zero. Finally, both thickness and depth maps for each projection view are saved into a single multi-channel image file.

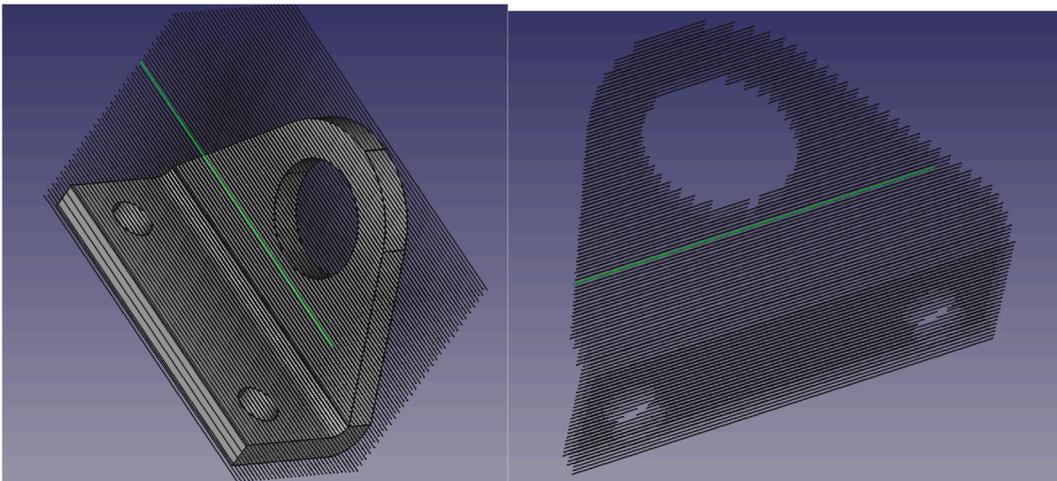


Fig. 3. Demonstration of thickness view generated from rays and the part; and intersected result by Boolean operation.

4.1.5. Watertight surface mesh preferred

In order to guarantee that the count of intersection vector is an even number to generate thickness, the surface mesh generated must be watertight and manifold. Watertight is a property to describe close surface mesh without holes [50]. Note that, not all meshing tool can convert a geometry file into a triangulation mesh with watertight property.

Fig. 4 (a) shows a meshed part that is not water-tight. There are 4 crevices on the curved edge on the left side. Considering the part as a closed container full of water, leakage happens at these crevices. The result of thickness view generation on the non-water-tight mesh is two unexpected dark pixels, see the leftmost part of Fig. 4 (b). Since some rays, shown as black points, may fall in the crevices and no intersection point is captured, the algorithm described above will fail for the odd number of the vector element count. Those empty pixels represent tiny holes, erroneous topology feature. Therefore, an interpolation strategy is used to fill the false pixel with nearby the average pixels. This interpolation may not always work. A threshold of 3 % erroneous pixel count is adopted, this sample is excluded if the threshold is surpassed.

Besides watertightness, input mesh should contain only exterior facets but not interior facets, otherwise, an odd number of intersection count leads to failure of thickness calculation. No interior facets will be guaranteed if a manifold geometry is created from CAD software, if interior face is not kept during Boolean operation. This thickness view generation algorithm applies to either single part, or an assembly of several parts (either in contact or merged into a single part by Boolean fusion), of which each part has watertight property.

5. Neural network design

Fig. 5 shows the schematic of the proposed model, coined as DTV-CNN, which is composed of a convolution neural network (CNN) as the primary submodel for depth-thickness-view and a multiple linear parameter (MLP) as the secondary submodel. According to literature review, multi-submodel can improve accuracy and robustness. For instance, it is challenging for a visual representation model to distinguish a toy airplane from a real airplane, without considering scalar geometric properties like characteristic length and volume.

5.1. CNN submodel

The minimum input view count is 3, which is a common practice to have at least 3 views in any mechanical drawing. Selecting only one view by max-pooling may lead to loss of information, instead, image concatenating is used in this CNN submodel to retain essential information. Data augmentation techniques such as flipping and rotation has been incorporated. A shallow neural network with only 3 Conv2D layers is employed, which is the key for the low training computation cost.

5.2. Multiple linear parameters (MLP) submodel

Only several geometry properties inexpensive to calculated in the preprocessing stage were selected, because MLP is only assistive while CNN is the primary model to extract topology information from geometry. More complex geometrical parameters are not introduced for its extra computation cost in the preprocessing stage.

In this paper, 6 geometrical parameters, independent of spatial position, i.e. translation and rotation, are selected. Those parameters are surface area and volume for the shape, the maximum length, 2 length aspect ratios and volume for the oriented bounding box.

For 3D geometry, area and volume are scaled into same unit as the maximum length for the oriented bounding box.

$$S_{scaled} = \sqrt[3]{S} \quad (1)$$

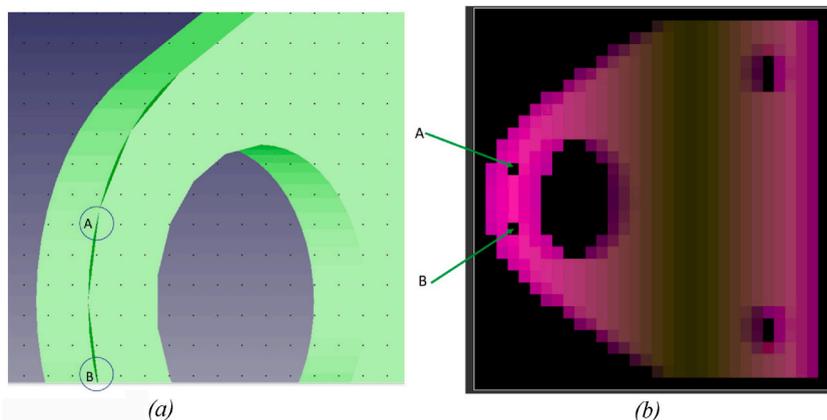


Fig. 4. Impact of (a) non-water-tight mesh on (b) thickness and depth views image quality.

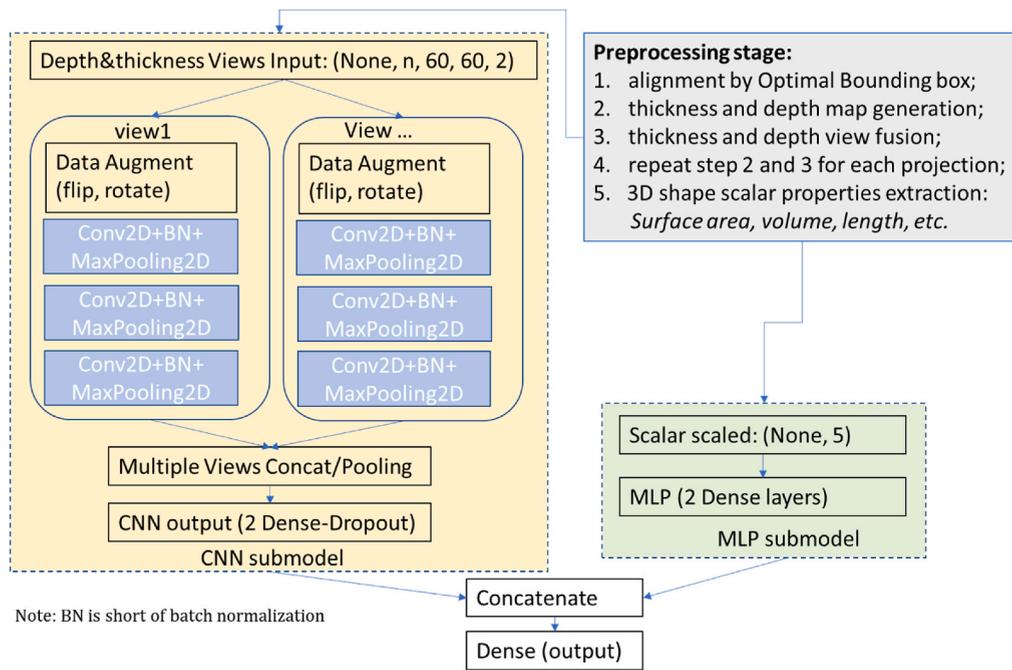


Fig. 5. Schematic of 3DTV-CNN model for 3D part classification.

and

$$V_{scaled} = \sqrt[3]{V} \quad (2)$$

where S is surface area (m^2); and V is volume (m^3)

Scalars are further normalized into the unity range of [0, 1], before feeding into MLP.

6. Evaluation and benchmarking

The software platform for this study is Python 3.7.9 and TensorFlow-2.5 on Windows 10 64bit. The source code is available at github <https://github.com/AICAE/3DPartClassification>, and the preprocessed image datasets in numpy file format are inside the data subfolder. Deep learning models are benchmarked on an Intel (R) Core (TM) i5-6200U 2 cores 4 thread laptop CPU with a maximum frequency 2.4 GHz, without any GPU acceleration. During the model benchmarking, the prepressing stage, i.e. transforming 3D geometry either as mesh format or CAD exchange format into image, is excluded. The time to preprocess each geometry sample file is less than 1 s using single thread; while the time spent on pre-processing stage can be accelerated by parallel computation approach like multi-threading.

6.1. Comparison with MVCNN, VoxNet and PointNet

This proposed DTV-CNN model has been benchmarked against reported 3D geometry classification models such as MVCNN, PointNet and VoxNet. The benchmark results on the ModelNet10 and ModelNet40 datasets are presented in Table 3. This comparison is limited to models which have source code available in public source code repository, therefore, the training time can be benchmarked against the same CPU. Source codes for MVCNN and VoxNet have been ported to use TensorFlow V2 APIs, available on github

Table 3

Comparing performance of DTV-CNN with MVCNN and VoxNet.

model variants	ModelNet10 accuracy	ModelNet40 accuracy	Trainable Parameters	Training time per epoch (s)	GFLOPs	Sample image size (byte)
3DTV-CNN	92 %	88 %	2,684,506	60	0.006	$3 \times 60 \times 60 \times 2$
6DTV-CNN	94 %	89.2 %	5,163,290	118	0.011	$6 \times 60 \times 60 \times 2$
MVCNN-12	NA	89.9 %	58,445,224	1800	0.28	$12 \times 227 \times 227 \times 3$
VoxNet	92 %	83 %	764,264	240	unavailable	$32 \times 32 \times 32$
PointNet	NA	89 %	3,480,049	unavailable	0.45	$4096 \times 3 \times 4$

(<https://github.com/AICAE/>). Note that, recent models like MVCNN-MultiRes [10] has already attained a higher accuracy of 93.6 %, with the cost of 20 views.

For the ModelNet10 mesh-based dataset, some samples fail to generate thickness view due to the lack of manifold/watertight shape characteristics. Interpolation from nearby pixel is employed to cope with this situation, if the percentage of interpolated pixel count is less than 3 %. 3808 out of ModelNet10 dataset of 4899 models are useable to benchmark our model. The training time per epoch in Table 3 is normalized for the total 4000 samples.

The image size in byte is formulated as

$$S_{im} = n \times W \times H \times D \quad (3)$$

where n is view count; W is image width in pixel, H is image height in pixel; D is channel depth in byte per pixel.

The selection of channel count is 2 or 3 in this study, and the input data can be treated as a normal RGB image to reuse image convolutional operator. If a future dataset contains color information, in the datasets, a special CNN operator for image with 6 channels would be straight-forward to design.

The advantage of DTV-CNN is apparent, in both accuracy and speed. As a 2D image input, the resolution is much lower than the MVCNN model, meaning a dramatical reduction in both computation cost and model parameter count. MVCNN with 12 views takes 30 minutes for each training epoch for ModelNet40 dataset, while it takes only 2 minutes for 6 view DTV-CNN. Despites 22 % of total samples are excluded due to the failure of depth and/or thickness view generation, the training speed of DTV-CNN is one order of magnitude faster.

Comparing the input image data size, this 3 view DTV-CNN has the similar byte size as VoxNet and accuracy for ModelNet10 dataset. Input data size for each sample is 21600 bytes for 3 view DTV-CNN; while a 3D convolution operator is more time-consuming than a 2D convolutional operator. For example, the training time per epoch, 60 s, is just 25 % of VoxNet. Not to mention, 3 view DTV-CNN has a better accuracy than VoxNet for ModelNet40 dataset.

In order to exclude the impact of dynamic frequency scaling on a mobile CPU, computational cost to train each image in one epoch is given as GFlops in Table 3 by Tensorflow's profiling API. Given that the point count is 4096 and single-precision float point is used as coordinate values, the computational cost of PointNet is about 2 orders of magnitude higher than DTV-CNN, while the accuracy and parameter is comparable. Therefore, it is confirmed DTV-CNN is an efficient model for 3D shape analysis.

6.2. Ablation study of model variants

Performance of model variants on ModelNet10 and FreeCAD library datasets aligns with findings on ModelNet40 dataset. Table 4 illustrates the accuracy of different model variants on these two datasets. The baseline model uses 3 depth-thickness projection views; Other model variants such as doubling the view count, and using only either thickness or depth channel are studied to identify the key enablers.

The coupling of depth map and thickness map as a dual-channel image is found to be the key to high accuracy. A single channel image of either thickness or depth map, does not reduce the training time as with dual-channel image input, but the difference in classification accuracy is significant for ModelNet10 and FreeCAD library dataset. The volumetric density approach [21] has a similar information presentation style as thickness view in this paper, but depth and thickness information are not bound together before image convolution. Emphasis on information locality has shown its advantage regarding training speed; DTV-CNN uses several hours on a mobile CPU for ModelNet10 dataset, instead of "22 h on a 3.60 GHz Intel i7-4790 CPU and an NVIDIA TitanX (Pascal) GPU". In particularly, DTV-CNN outperforms by 20 % and 33 % thickness-only or depth-only views, for complicate 3D shape in ModelNet10 dataset. The numerical experiment confirms the fusion of depth and thickness map which improves the information locality is crucial for CNN on 3D datasets.

The MLP submodel can improve the classification accuracy by 4 % without an significant increase of training time and model parameter count. For example, the volume of the shape helps to distinguish a real aircraft from its toy counterpart. Therefore, the MLP submodel is worth of adoption to stabilize the CNN submodel. On the other hand, doubling the view count to 6 can improve the accuracy by 2 %, but the training time as well as the total trainable parameters increase proportionally. As recent researchers have stated, increasing the view count further will not help observably. A compromise of accuracy and speed could be reached for the applications on mobile platforms and/or web browsers, where CNN on 3D dataset is barely reported. In addition, different pooling strategies have been evaluated. Maximum view pooling, instead of concatenating views as in the baseline model, this does not reduce the training time observably although the model parameters drop to 40 %.

Table 4
Performance of 3DTV-CNN model variants on ModelNet10 dataset and FreeCAD dataset.

model variants	ModelNet10 dataset	FreeCAD dataset	Trainable parameter count	Sample image size (byte)
3 DTV-CNN (baseline)	92 %	97 %	2,683,834	3 × 60 × 60×2
6 DTV-CNN	94 %	97 %	5,163,290	6 × 60 × 60×2
Only 3 thickness views	72 %	95 %	2,683,402	3 × 60 × 60
Only 3 depth views	59 %	90 %	2,683,402	3 × 60 × 60
3 DTV, no MLP	88 %	95 %	2,613,850	3 × 60 × 60×2
3DTV, max view pooling	90 %	97 %	1,078,234	3 × 60 × 60×2

Table 5
Performance of DTV-CNN on different datasets.

dataset	validation accuracy
FreeCAD-library	97 %
KiCAD-library	95 %
ModelNet10	92 %
ModelNet40	88 %
ShapeNetCore	87 % (see note)

Note: higher input image resolution and extra Con2D layers are used for ShapeNetCore dataset to improve accuracy.

6.3. Performance on different datasets

Table 5 is a summary of the performance of DTV-CNN for 3D part/shape classification. The new datasets introduced in this work, FreeCAD and KiCAD standard part libraries, have higher classification accuracy than ModelNet10 and ModelNet40. Samples from both CAD datasets are comparatively simple components, not a complex product as in ModelNet. Meanwhile, this could be partially attributed to better depth and thickness view image quality, for its CAD data format with water-tight and manifold nature.

Our thickness view projection algorithm can not generate the perfect thickness map view for non-manifold and non-watertight shapes from ModelNet and ShapeNetCore datasets. For some regions where thickness can not be calculated for the existence of interior mesh facet (non-manifold) or mesh crevice (non-watertight), data thickness is obtained by extrapolation from nearby pixels. This introduced noise undermines the performance of DTV-CNN.

Finally, this DTV-CNN model has the potential to be deployed in edge computation scenario. If the trained model has already loaded into memory, a classification task for a new 3D shape takes 3 s on a low-end laptop CPU (Intel (R) Core (TM) i5-6200U 2 cores), during that period, 2 s spent on preprocessing can be further reduced by multi-threading.

7. Conclusion

In this paper, a 2.5D thickness projection view algorithm is proposed to accelerate 3D shape classification using a model of mixed CNN and MLP. This minimized input data size (3 views instead of 12 views or more for the MVCNN model) and neural network model parameter count, can significantly reduce the computation cost of model training, and contribute to the prosperity of machine learning on 3D shape.

This proposed model, DTV-CNN, has an apparent advantage in training and evaluation speed, although the accuracy is not the highest. A reasonable accuracy of 92 % on mesh dataset ModelNet10 is achieved using only 3 orthogonal views, while the training time is reduced to about tens of minutes on a laptop CPU. For realistic CAD shape datasets created in this study, the standard part library from FreeCAD and KiCAD, the validation accuracy can reach 97 % and 95 % respectively. The smaller model parameter count and faster training speed are due to the effective 3D topology information extraction from 2.5D thickness project views, and information locality by fusion of depth and thickness view. The classification model is further assisted by a MLP submodel of several key scalar geometrical properties. The introduction of scalar geometrical properties has ignorable computation cost but can lead to an observable accuracy improvement.

The small model size and training cost of this proposed 3D classification model can be integrated into CAD and CAE software, and serve as a key enabler for automated and intelligent engineering design in the future. Meanwhile, the real-time performance of this DTV-CNN model will broaden the application of machine learning on 3D shape to more resource-limited platforms.

Limitation and future work

In this paper, information locality has been improved by interweaving depth information into a conventional 2D projection view, while locality could be further improved by a special convolution operator, to correlate orthogonal XY-YZ-ZX views. For example, PointView-GCN with multi-level GCNs can hierarchically collect the shape features of single-view and their multi-view relations, therefore achieved a high accuracy of 95 %. If color information is available, the color channel should be interwoven with depth and thickness channel from the same projection to improve information locality.

As the category count increases in a real-world application, extra convolution layers and higher input image resolution are required to maintain the classification accuracy. Future work to deal with category scaling-up challenge would be multi-submodel approach. One principle of selection of submodel is orthogonality, if DTV-CNN is selected as the primary model and MLP and the geometrical scalar properties as the second MLP submodel, then topology-based model like NeRF or graph attention network could be more helpful than voxel or other visualization approaches. Moreover, transformer-based deep learning model could also be explored for 3D shape classification and generation.

Disclaimer and acknowledgement

This is personal research, not funded or related with employment. Thanks are owned to the author's family members, community of

FreeCAD open source community, and Dr Weidi Xie from the University of Oxford.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- [1] B. Starly, A. Angrish, A. Bharadwaj, MVCNN++: CAD model shape classification and retrieval using multi-view convolutional neural networks, *J. Comput. Inf. Sci. Eng.* 21 (1) (2020) 1–27, <https://doi.org/10.1115/1.4047486>.
- [2] A.A. Soltani, et al., Synthesizing 3D shapes via modeling multi-view depth maps and silhouettes with deep generative networks, in: *IEEE Conference on Computer Vision & Pattern Recognition*, 2017, <https://doi.org/10.1109/CVPR.2017.269>.
- [3] B. Song, et al., GIFT: a real-time and scalable 3D shape search engine, in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2016, <https://doi.org/10.1109/CVPR.2016.543>.
- [4] Z. Zhang, P. Jaiswal, R. Rai, FeatureNet: machining feature recognition based on 3D convolution neural network, *Comput. Aided Des.* 101 (2018) 12–22, <https://doi.org/10.1016/j.cad.2018.03.006>.
- [5] A.R. Colligan, et al., Hierarchical CADNet: learning from B-reps for machining feature recognition, *Comput. Aided Des.* 147 (2022), 103226, <https://doi.org/10.1016/j.cad.2022.103226>.
- [6] L.d.F. Costa, R.M. Cesar, *Shape Classification and Analysis : Theory and Practice, second ed.*, CRC Press, 2009.
- [7] D. Maturana, S. Scherer, VoxNet: a 3D convolutional neural network for real-time object recognition, in: *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2015, <https://doi.org/10.1109/IROS.2015.7353481>.
- [8] S. Chen, et al., 3D point cloud processing and learning for autonomous driving: impacting map creation, localization, and perception, *IEEE Signal Process. Mag.* 38 (1) (2021) 68–86, <https://doi.org/10.1109/MSP.2020.2984780>.
- [9] A.X. Chang, et al., ShapeNet: an information-rich 3D model repository, *Computer Sci.* (2015), <https://doi.org/10.48550/arXiv.1512.03012>.
- [10] C.R. Qi, et al., Volumetric and multi-view CNNs for object classification on 3D data, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, <https://doi.org/10.1109/CVPR.2016.609>.
- [11] H. Su, et al., Multi-view convolutional neural networks for 3D shape recognition, in: *IEEE International Conference on Computer Vision, ICCV*, 2015, <https://doi.org/10.1109/ICCV.2015.114>.
- [12] M. Yavartanoo, E.Y. Kim, K.M. Lee Spnet, Deep 3D object classification and retrieval using stereographic projection, in: *Asian Conference on Computer Vision*, Springer, 2018, https://doi.org/10.1007/978-3-030-20873-8_44.
- [13] H. You, et al., PVRNet: point-view relation neural network for 3D shape recognition, in: *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, <https://doi.org/10.1609/aaai.v33i01.33019119>.
- [14] S. Qi, et al., Review of multi-view 3D object recognition methods based on deep learning, *Displays* 69 (2021), 102053, <https://doi.org/10.1016/j.displa.2021.102053>.
- [15] C. Ma, et al., Learning multi-view representation with LSTM for 3D shape recognition and retrieval, *IEEE Trans. Multimed.* (2019) 1169–1182, <https://doi.org/10.1109/TMM.2018.2875512>.
- [16] C.R. Qi, et al., Volumetric and multi-view CNNs for object classification on 3D data, in: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, <https://doi.org/10.1109/CVPR.2016.609>.
- [17] Y. Tan, J. Meng, J. Yuan, Multi-view harmonized bilinear network for 3D object recognition, in: *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, <https://doi.org/10.1109/CVPR.2018.00027>.
- [18] Q. Yu, et al., Latent-MVCNN: 3D shape recognition using multiple views from pre-defined or random viewpoints, *Neural Process. Lett.* 52 (1) (2020) 581–602, <https://doi.org/10.1007/s11063-020-10268-x>.
- [19] X.F. Han, H. Laga, M. Bennamoun, Image-based 3D object reconstruction: state-of-the-art and trends in the deep learning era, *IEEE Trans. Pattern Anal. Mach. Intell.* (2021), <https://doi.org/10.1109/TPAMI.2019.2954885>.
- [20] P. Zanuttigh, L. Minto, Deep learning for 3D shape classification from multiple depth maps, in: *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, <https://doi.org/10.1109/ICIP.2017.8296956>.
- [21] L. Minto, P. Zanuttigh, G. Pagnutti, Deep learning for 3D shape classification based on volumetric density and surface approximation clues, in: *International Conference on Computer Vision Theory & Applications*, 2018, <https://doi.org/10.5220/0006619103170324>.
- [22] J. Wang, Y. He, H. Tian, Voxel-based shape analysis and search of mechanical CAD-models, *Forsch. Im. Ingenieurwes.* 71 (3–4) (2007) 189–195, <https://doi.org/10.1007/s10010-007-0057-5>.
- [23] S. Kumawat, S. Raman, LP-3DCNN: unveiling local phase in 3D convolutional neural networks, in: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2019, <https://doi.org/10.1109/CVPR.2019.00504>.
- [24] H. Ye, Z. Du, F. Cao, A novel 3D shape classification algorithm: point-to-vector capsule network, *Neural Comput. Appl.* 33 (23) (2021) 16315–16328, <https://doi.org/10.1007/s00521-021-06231-z>.
- [25] C.Q. Huang, et al., Dual-graph attention convolution network for 3-D point cloud classification, *IEEE Transact. Neural Networks Learn. Syst.* (2022) 1–13, <https://doi.org/10.1109/TNNLS.2022.3162301>.
- [26] Y. Feng, et al., MeshNet: mesh neural network for 3D shape representation, in: *AAAI'19: AAAI Conference on Artificial Intelligence*, 2019, <https://doi.org/10.1609/aaai.v33i01.33018279>.
- [27] R. Hanocka, et al., MeshCNN: a network with an edge, *ACM Trans. Graph.* 38 (4) (2019), <https://doi.org/10.1145/3306346.3322959>. Article 90.
- [28] A. Kaiser, J.A. Ybanez Zepeda, T. Boubekeur, A survey of simple geometric primitives detection methods for captured 3D data, *Comput. Graph. Forum* (2018), <https://doi.org/10.1111/cgf.13451>.
- [29] M. Rucco, et al., A methodology for part classification with supervised machine learning, *AI EDAM (Artif. Intell. Eng. Des. Anal. Manuf.)* 33 (1) (2019) 100–113, <https://doi.org/10.1017/S0890060418000197>.
- [30] B. Jiang, et al., Semi-Supervised learning with graph learning-convolutional networks, in: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, <https://doi.org/10.1109/CVPR.2019.01157>.
- [31] N. Verma, E. Boyer, J. Verbeek, FeaStNet: feature-steered graph convolutions for 3D shape analysis, in: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, <https://doi.org/10.1109/CVPR.2018.00275>.
- [32] S.S. Mohammadi, Y. Wang, A.D. Bue, Pointview-Gcn, 3D shape classification with multi-view point clouds, in: *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, <https://doi.org/10.1109/ICIP42928.2021.9506426>.
- [33] X. Wei, R. Yu, J. Sun, View-GCN: view-based graph convolutional network for 3D shape analysis, in: *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, <https://doi.org/10.1109/CVPR42600.2020.00192>.
- [34] X. Wei, R. Yu, J. Sun, Learning view-based graph convolutional network for multi-view 3D shape analysis, *IEEE Trans. Pattern Anal. Mach. Intell.* 45 (6) (2023) 7525–7541, <https://doi.org/10.1109/TPAMI.2022.3221785>.
- [35] H. You, et al., PVNet: a joint convolutional network of point cloud and multi-view for 3D shape recognition, in: *Proceedings of the 26th ACM International Conference on Multimedia*, Association for Computing Machinery, Seoul, Republic of Korea, 2018, pp. 1310–1318, <https://doi.org/10.1145/3240508.3240702>.

- [36] M. Lee, J. Kim, S.C. Yu, Robust 3D shape classification method using simulated multi view sonar images and convolutional neural network, in: OCEANS 2019 - Marseille, 2019, <https://doi.org/10.1109/OCEANSE.2019.8867438>.
- [37] H. Tabia, et al., Non-rigid 3D shape classification using bag-of-feature techniques, in: 2011 IEEE International Conference on Multimedia and Expo, 2011, <https://doi.org/10.1109/ICME.2011.6011970>.
- [38] M.A. Uy, et al., Revisiting point cloud classification: a new benchmark dataset and classification model on real-world data, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, <https://doi.org/10.1109/ICCV.2019.00167>.
- [39] W. Zhirong, et al., 3D ShapeNets: a deep representation for volumetric shapes, in: 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, <https://doi.org/10.1109/CVPR.2015.7298801>.
- [40] Q. Zhou, A. Jacobson, Thingi10K: A Dataset of 10,000 3D-Printing Models, 2016, <https://doi.org/10.48550/arXiv.1605.04797> arXiv.
- [41] K. Mo, et al., PartNet: a large-scale benchmark for fine-grained and hierarchical part-level 3D object understanding, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, <https://doi.org/10.1109/CVPR.2019.00100>.
- [42] S. Koch, et al., ABC: a big CAD model dataset for geometric deep learning, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, <https://doi.org/10.1109/CVPR.2019.00983>.
- [43] Z. Gao, H. Xue, S. Wan, Multiple Discrimination and Pairwise CNN for view-based 3D object retrieval, Neural Network. 125 (2020) 290–302, <https://doi.org/10.1016/j.neunet.2020.02.017>.
- [44] D.-Y. Chen, et al., On visual similarity based 3D model retrieval, Comput. Graph. Forum 22 (3) (2003) 223–232, <https://doi.org/10.1111/1467-8659.00669>.
- [45] A. Liu, et al., Multi-modal clique-graph matching for view-based 3D model retrieval, IEEE Trans. Image Process. 25 (5) (2016) 2103–2116, <https://doi.org/10.1109/TIP.2016.2540802>.
- [46] T. Wu, et al., OmniObject3D: large-vocabulary 3D object dataset for realistic perception, reconstruction and generation, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2023, <https://doi.org/10.1109/CVPR.2019.00983>.
- [47] K.L. Edwards, Standard handbook of machine design, Mater. Des. 17 (3) (1996) 175–176.
- [48] N. Sedaghat, et al., Orientation-boosted voxel nets for 3D object recognition, in: British Machine Vision Conference, 2017, <https://doi.org/10.48550/arXiv.1604.03351>, 2017.
- [49] A. Hamdi, et al., MVTN: multi-view transformation network for 3D shape recognition, in: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), virtual conference, 2021, <https://doi.org/10.48550/arXiv.2011.13244>.
- [50] L. Caraffa, M. Brédif, B. Vallet, 3D octree based watertight mesh generation from ubiquitous data, in: ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2015, pp. 613–617, <https://doi.org/10.5194/isprsarchives-XL-3-W3-613-2015>. XL-3/W3.