3D Model Classification and Retrieval Based on Multi-View Convolutional Neural Networks

Vedant Bommanahally
vbommanahally@ryerson.ca
Department of Electrical and Computer Engineering
Toronto Metropolitan University, Toronto, Canada.

***Abstract*—In recent years advancements in computer hardware has led to an increase of 3D shape data which requires a classification and retrieval system in order to be used effectively. A classification and retrieval system is capable of sorting 3D models into categories or return a ranked list of models that are most similar to a query. Difficulty in developing such a system stem from finding a consistent way of representing 3D geometry so that the data can be used to train a neural network. Previous work that has been done on this subject tend to use volumetric grid, multiple-viewpoint images, point clouds or polygonal meshes to represent the data. This paper aims at exploring multiple techniques that other researchers have used to solve the problem with a focus on using multiple-viewpoint images because this method has shown to be the most effective. Three main components within the system will be studied closely including the representation of data, neural network architecture and the similarity measure. One or a combination of the techniques studied will be implemented and tested on commonly used datasets for this application. It is important to address this problem because progress in this area of research can have major implications on many emerging technologies such as augmented and virtual reality, autonomous driving and 3D printing.**

***Index Terms*—volumetric grid, multiple-viewpoint images, point clouds, polygonal meshes, similarity measure, augmented reality, virtual reality, 3D printing.**

I. Introduction

The advancement in computer hardware during the last few years has led to an explosion in the use of 3D technology in fields such as augmented reality, virtual reality or computer vision. A particularly exiting application of computer vision is autonomous driving. However, in order to apply computer vision to autonomous driving a machine learning algorithm must be created which can classify 3D models to differentiate between people and cars on the road [1]. A lot of progress has been made in 2D image recognition using CNN however there are many challenges with applying CNN for 3D model classification. For instance, representing a 3D model as a 2D manifold is not like 2D image representation and there is no standardized way of storing 3D geometric shapes as data [2]. 3D models can be represented in many formats before being fed into a neural network such as volumetric grid, multiple-viewpoint images, point clouds or polygonal meshes [3]. It is challenging to use polygonal meshes as inputs to neural networks because each model in the dataset will have a different triangle count as opposed to a dataset of 2D images where all elements share the same resolution and pixel dimension [3]. Due to the advent of 3D printing many 3D models are being generated using various tools such as 3D printers and uploaded to the internet. Instead of creating a 3D model from scratch it is much more efficient to download a similar model from the internet and incrementally adjust which can reduce production cost and shorten the design time. However, finding the desired model from the large pool of models on the internet is a difficult task and requires a machine learning based 3D model retrieval system. For the various applications given above, it is clear that a machine learning based 3D model classification and retrieval system is an important area of research to meet the demands of the future.

II. Literature Review

The paper proposed by Ding et al. [4] describes a method for 3D model retrieval that uses 2D projective views to train a CNN. A representative view is chosen by using the K-means algorithm and the ResNet50 CNN model is used to find the feature vector for the model. Euclidean distance between the feature vectors of different models is used as the similarity measure. The paper proposed by Angrish et al. [5] improves upon the previous method by processing part dimension data along with the 3D models to build a retrieval system suitable for engineering CAD models. Twelve parallel and pretrained ResNet50 CNN models are used to find the feature vectors of 12 projective views. The feature vectors are aggregated by finding the element-wise maximum and the bounding box dimensions of the CAD model are concatenated to this vector. Cosine distance is used as the similarity measure. The paper proposed by Taybi at al. [6] takes a novel approach to this problem by using 2D cross-sections of the 3D models instead of projective views. 2D cross-sections are found by taking the intersection between the mesh and 64 planes placed along the three axes. All 64 slices are stacked together and fed into a CNN to extract the feature vector of the mesh. Models in the same class may not have similar images from the same camera location due to inconsistent orientation of the models. The previous projective view based approaches use 12 parallel CNNs and take the element-wise maximum of their outputs but this results in information loss. The work proposed by Chen et al. [7] attempts to deal with this issue by only fusing the outputs of the 2nd and 4th CNNs. The paper proposed by Sfikas et al. [8] describes a method for

3D model retrieval where panoramic images are used as the input to the CNN. Panoramic images are obtained by placing a cylinder around the model and projecting rays from the model onto the inside surface of the cylinder. Each model will be represented by a 3-channel panoramic image which will be computed for the three axes. The ensemble of panoramic images is input into 3 parallel CNNs followed by 2 fully connected layers. Outputs of the CNN are averaged together to get the descriptor for the model and the Manhattan distance is used as the similarity measure. The work proposed by Ding et al. [9] aims to perform 3D classification given a single view of a 3D shape. For training, images of the model from 6 sides are taken and fed into 6 parallel CNN models sharing the same convolutional and pooling layers but having their own fully connected and SoftMax layers. During classification a representative image with the highest surface complexity is selected and input into all 6 CNNs and some statistical criteria is used to determine which class the model belongs to. One of the drawbacks of the multi-view based approach is that it does not consider the internal structure of the models. Gomez-Donoso et al. [10] attempt to solve this issue by projecting point cloud data onto three orthogonal planes that are placed at the center of the model. This produces 3 2D images per sample which are subsequently fed into 3 parallel GoogLeNets followed by a concatenation layer and a fully connected layer which classifies the model. Which similarity measure to use to retrieve models that are most like the query is an important aspect to consider. The research conducted by Watanabe et al. [11] shines light on this issue because they have tested 4 similarity measures for 3D model retrieval. This includes the Euclidean distance, Pearson product-moment correlation coefficient, cosine similarity and weighted Jaccard coefficient. The research done by Ma et al. [12] attempt to build on the multi-view CNN model by incorporating LSTM networks into the system by treating different views of an object as a temporal sequence. The network architecture is composed of a 2D CNN network, a highway network, a two-layer bidirectional LSTM, a sequence voting layer and a fully connected layer. The paper written by Zhou et al. [13] discusses a method of projecting the point cloud representation of a 3D model onto the inside surface of a sphere to capture the internal structure of the models. A uniform distribution of points on all surfaces of the model are created and converted into spherical coordinates from rectangular coordinates. A 2D mapping is created by dividing the inner surface of the sphere into rectangular regions where the center of the region represents the x and y coordinates while the value of the mapping is the maximum distance of all points in the region.

III. Problem Statement

The availability of hardware that rely on 3D shape data, such as 3D scanners and printers, has dramatically increased in the past few years. To accommodate the massive flood of 3D shape data researchers must work on creating an optimized system for the classification and retrieval of 3D models. Such a system will have an impact on various technologies and industries such as autonomous driving, virtual and augmented reality, 3D printing and manufacturing. Ignoring this problem will have consequences for people working in these industries, especially manufacturing because they will need to spend more time looking for a CAD file that suits their needs. Although researchers have come up with various ways of tackling this problem there is still room for improvement. This research project aims at developing a classification and retrieval system that optimizes either one or all the following criteria: accuracy, retrieval time, memory requirement or training time. This will be achieved by using a neural network for not only classifying the 3D models but also finding a descriptor of the model which can easily be compared with others. Various components of the system will be experimented on to find the combination that will yield the best performance. Some components that can be experimented on include the representation of 3D models as 2D images, the architecture of the neural network model and the similarity measure to compare one model to another. For this research paper the multi-view-based approach will be used because it has achieved state of the art results on the most used datasets.

IV. Proposed Methodology

The first step in the procedure is to scale the model to fit inside of a sphere with a radius of 100 units. This is followed by placing 64 cameras at different locations on a sphere that has a radius of 200 units. The cameras are placed by using the formulas shown below where n and m represent the number of segments along the polar and azimuthal axis. To obtain 64 images the number of segments will have to be $m = 8$ and $n = 8$. The θ and ϕ variables represent the polar and azimuthal angles.

$$camPosX = 200\cos\left(\frac{2\pi\theta}{n}\right)\sin\left(\frac{\pi\phi}{m}\right)$$

$$camPosY = 200\sin\left(\frac{2\pi\theta}{n}\right)\sin\left(\frac{\pi\phi}{m}\right)$$

$$camPosZ = 200\cos\left(\frac{\pi\phi}{m}\right)$$

After the cameras are placed, the images were taken using a virtual camera pointed at the center of the model using directional lights placed along the x, y and z axis to evenly light the model. Some views are more suitable for classification and retrieval than other views. To determine the representative view for the dataset all 64 images will be fed into a CNN and their classification performance will be measured. The view with the highest accuracy will be chosen as the representative view. The architecture of the CNN used for this application is loosely based on the AlexNet architecture. It is composed of 5 convolutional layers where the first, second and fifth are followed by a max-pooling layer. The first convolutional layer uses 96 filters with a kernel size of 11×11 and stride size of 4×4. The second convolutional layer uses 256 filters with a kernel size of 5×5 and stride size of 4×4. The third and fourth convolutional layers use 384 filters with a kernel size of 3×3 and a stride size of 1×1. The last convolutional layer uses 256 filters with a kernel size of 3×3 and a stride size of 1×1. All the max-pooling layers use a pool size of 3×3 and a stride size of 2×2. A flatten layer is used to flatten the output tensor of the final max-pooling layer into a one-dimensional vector. This is followed by 2 fully connected layers with 4096 neurons with dropout layers and another fully connected layer with 10 neurons. The final layer must have 10 neurons because the dataset consists of 10 classes. The structure of the CNN is described in more detail in Table 1.

Table 1: The structure of the CNN

Layer name	Output size	Layer information
Input layer	$224 \times 224 \times 3$	NA
Convolutional layer 1	$56 \times 56 \times 96$	$filters = 96$ $kernel\_size = 11 \times 11$ $strides = 4 \times 4$
Batch normalization	NA	NA
Maxpooling layer 1	$28 \times 28 \times 96$	$pool\_size = 3 \times 3$ $strides = 2 \times 2$
Convolutional layer 2	$7 \times 7 \times 256$	$filters = 256$ $kernel\_size = 5 \times 5$ $strides = 2 \times 2$
Batch normalization	NA	NA
Maxpooling layer 2	$4 \times 4 \times 256$	$pool\_size = 3 \times 3$ $strides = 2 \times 2$
Convolutional layer 3	$4 \times 4 \times 384$	$filters = 384$ $kernel\_size = 3 \times 3$ $strides = 1 \times 1$
Batch normalization	NA	NA
Convolutional layer 4	$4 \times 4 \times 384$	$filters = 384$ $kernel\_size = 3 \times 3$ $strides = 1 \times 1$
Batch normalization	NA	NA
Convolutional layer 5	$4 \times 4 \times 256$	$filters = 256$ $kernel\_size = 3 \times 3$ $strides = 1 \times 1$
Batch normalization	NA	NA
Maxpooling layer 3	$2 \times 2 \times 256$	$pool\_size = 3 \times 3$ $strides = 2 \times 2$
Flatten layer	1024	NA
Dense layer 1	4096	NA
Dropout layer 1	NA	NA
Dense layer 2	4096	NA
Dropout layer 2	NA	NA
Dense layer 3	10	NA

To perform model retrieval, input the representative view of the query model into the neural network and classify the model into one of the categories. Use the output of the second to last fully connected layer (Dense layer 2) as the descriptor for the query. This descriptor will be compared with the descriptors of all other models in the dataset to retrieve the models that are most similar to the query. A similarity measure is required to quantify the similarity between the query and the other models. The similarity measures that were tested for this application include the Euclidean distance, the cosine distance and the Manhattan distance. If $\vec{x}$ and $\vec{y}$ are two vectors in two-dimensional space, then the formulas to calculate the Euclidean distance, Manhattan distance and Cosine distance are shown below.

$$EuclideanDistance = \sqrt{(x_1 - y_1)^2 + (x_2 - y_2)^2}$$

$$ManhattanDistance = |x_1 - y_1| + |x_2 - y_2|$$

$$CosineDistance = 1 - \cos\theta = 1 - \frac{\vec{x} \cdot \vec{y}}{\|x\| \|y\|}$$

V. Experimental Setup and Results

All experiments are conducted on a Ryzen 5 3550H PC with 8 GB of ram and without the use of a GPU. The ModelNet10 dataset is used and contains 4899 3D models divided into 10 categories. The categories include “bathtub”, “bed”, “chair”, “desk”, “dresser”, “monitor”, “nightstand”, “sofa”, “table” and “toilet”. The models in the dataset are stored in the OFF file format which essentially lists the coordinates of all vertices in a model followed by the triangles in the model. A program developed in the Processing programming environment is used to render the OFF files and capture images of the model from various view points and lighting conditions. Example images from the dataset are shown in Figure 1 which depicts a chair model from multiple viewpoints. The number of samples belonging to each category is shown in Table 2.

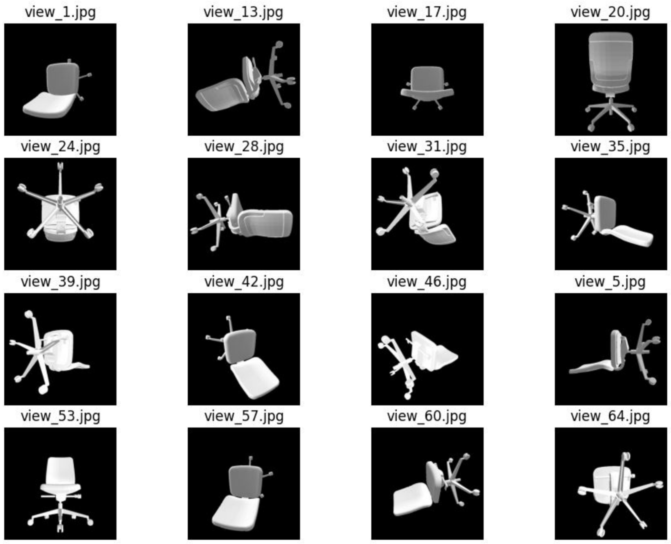


Fig. 1. Multiple views of chair model.

Table 1: Samples per category in ModelNet10

Category	Number of samples
bathtub	156
bed	615
chair	989
desk	286
dresser	286
monitor	565
night stand	286
sofa	780
table	492
toilet	444

The dataset is split into a training, validation and testing set consisting of 3920, 490 and 489 samples respectively. The classification performance of the neural network is determined by measuring the accuracy of the neural network over the testing set. The retrieval performance of the neural network is measured by first finding the descriptor of each model in the training set. This is followed by finding the descriptor of the query, which is taken from the testing set, and using the chosen similarity measure to retrieve the 100 most similar models from the training set. Retrieval accuracy is measured as the percentage of retrieved models that are in the same category as the query model. In order to determine which view to use in the neural network all 64 views were passed through the same CNN and the view with the highest accuracy will be selected. The Adam optimizer and sparse categorical cross entropy is being used. The CNN is trained using a learning rate of 0.0001 over 5 epochs and the dataset is split into batches of 128. Figure 2 shows the results from this initial run of the CNN where the most promising views appear to be view numbers 3 and 47 which achieved an accuracy of 81.39% and 78.94%. After further experimentation view number 47 is chosen as the representative view for the dataset.

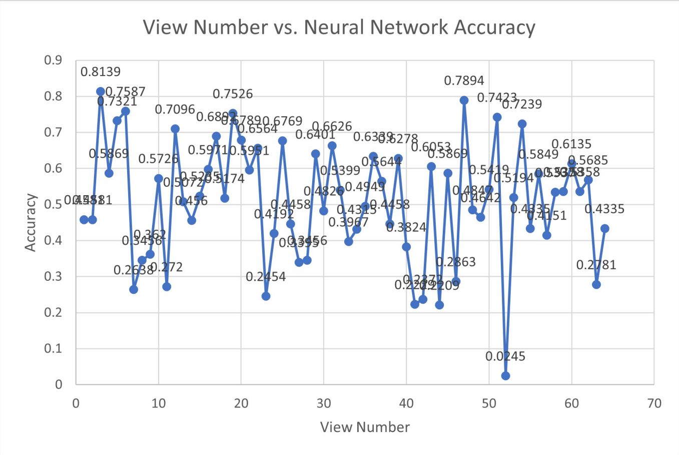
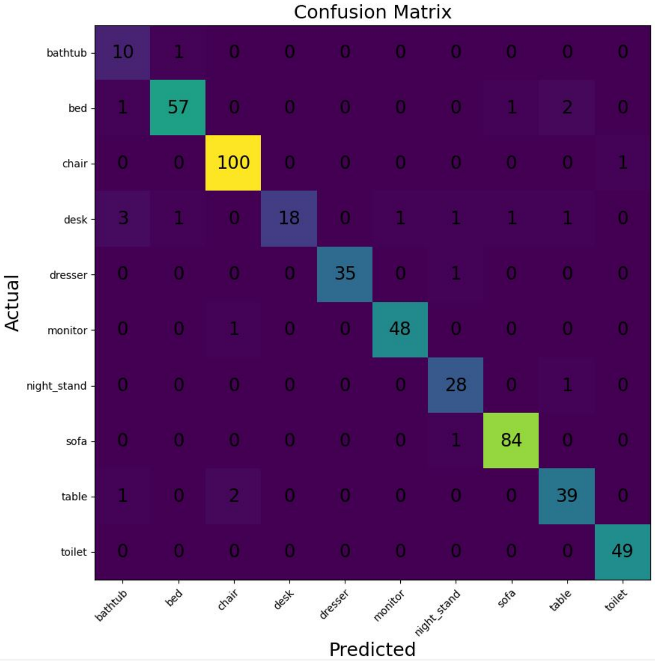


Fig. 2. View Number vs. Neural Network Accuracy.

The CNN is trained again but with a learning rate of 0.00001 over 20 epochs in order to acheive the highest accuracy possible. This resulted in an accuracy of 95.71% when the neural network is evaluated on the testing dataset. The confusion matrix is shown in Figure 3 and since the largest numbers are on the diagonal of the matrix it shows that the classification performance is very good for all categories.



	bathtub	bed	chair	desk	dresser	monitor	night_stand	sofa	table	toilet
bathtub	10	1	0	0	0	0	0	0	0	0
bed	1	57	0	0	0	0	0	1	2	0
chair	0	0	100	0	0	0	0	0	0	1
desk	3	1	0	18	0	1	1	1	1	0
dresser	0	0	0	0	35	0	1	0	0	0
monitor	0	0	1	0	0	48	0	0	0	0
night_stand	0	0	0	0	0	0	28	0	1	0
sofa	0	0	0	0	0	0	1	84	0	0
table	1	0	2	0	0	0	0	0	39	0
toilet	0	0	0	0	0	0	0	0	0	49

Fig. 3. Confusion matrix from CNN evaluated on the testing dataset.

The training and validation accuracy and loss are also plotted over the number of epochs in Figure 4 to show that there is very little overfitting which means the neural network will be able to perform at the same level when it encounters samples that it has never seen before.

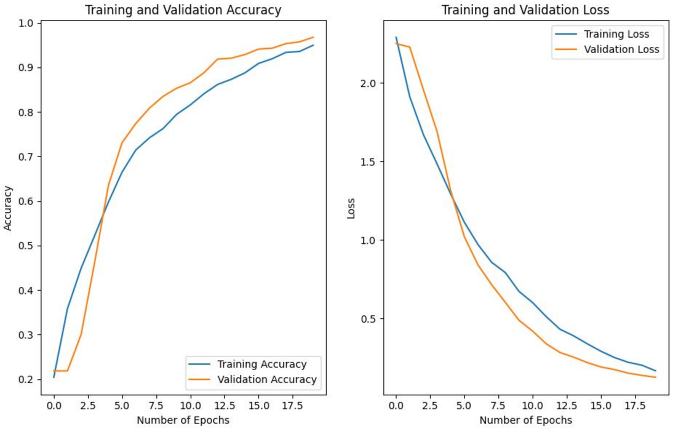


Fig. 4. Training and validation accuracy and loss plotted over the number of epochs.

The descriptor of a model is found by taking the output of the second to last layer in the CNN after passing the representative view of the model through the CNN. In order to measure the retrieval performance of this scheme the descriptors of all models in the training and testing datasets are found and stored in an array. All models in the testing dataset are used as a query to the models in the training dataset. For each query 100 of the most similar models, determined by the similarity measure, are retrieved and the number of models that belong to the same category as the query model are counted. The retrieval accuracy is defined as the number of retrieved models that share the same category as the query. The average retrieval accuracy is calculated over all models in the testing dataset. This procedure will be carried out once for each of the similarity measures described above and the average retrieval accuracy is compared to find the optimal similarity measure. The results of this experiment are shown in Table 2. The results show that the retrieval performance does not depend on the similarity measure although the cosine distance slightly out performed the other measures.

Table 2: Average retrieval accuracy obtained from each similarity measure

Similarity measure	Average retrieval accuracy
Eulclidean Distance	84.64%
Manhattan Distance	85.84%
Cosine Distance	85.90%

Examples of the retrieval results are shown in Figures 5, 6, and 7. The results show that the neural network is able to retrieve models that are not only in the same category as the query model but also containing the same characteristics as the query model. For instance, in Figure 5 the query model is a chair which has a pair of U-shaped legs and many of the returned models also have U-shaped legs. In addition, the query chair has very thin legs and a sqare shaped seat and most of the retrieved models also have these characteristics. In Figure 6, the query model is a rectangular table and the neural network is able to recognize that the rectangular tables are more similar to the query than the circular tables since the rectangular tables are ranked higher than the circular tables. Although the performance of the neural network is good for some categories there are other categories where the retrieval performance is lacking. For example, in Figure 7 the query model is a desk and only the top 3 retrieved models belong to the desk category. The neural network is confusing models belonging to other categories such as table, bed and nightstand for a desk. This is happening because the table, bed and nightstand models look very similar at the angle that the images are taken from.

VI. Conclusion

The advancements in 3D technologies in many fields has led to the urgent need of a classification and retrieval system optimized for 3D models. In this paper a 3D model classification and retrieval scheme is proposed that attempts to optimize one of the following criteria: accuracy, retrieval time, memory requirement or training time. The proposed scheme utilizes many images of the same model taken at different angles and attempts to find the viewpoint that carries the most information in the dataset and trains a neural network using images taken from that viewpoint only. The CNN that will be trained is based off AlexNet and will be used to perform model classification directly. Model retrieval will be done by extracting the output of the second to last fully connected layer in the CNN and using it as a descriptor for the model. A similarity measure will be used to compare the descriptor of the query model to the descriptors of the models in the database. The impact of the choosen similarity measure on retrieval performance is also studied. The experimental results show that the proposed scheme is a viable method for the classification and retrieval of 3D models. The classification accuracy of the proposed scheme is very high at 95.71% which is competitive with the state of the art models discussed in the literature review section. The retrieval performance of the proposed scheme is acceptable at around 85.90% and the neural network has shown the capability to ascertain the characteristics of models in the same category. The experiments also show that the similarity measure used in the retrieval process does not have a large impact on performace. A major advantage of this

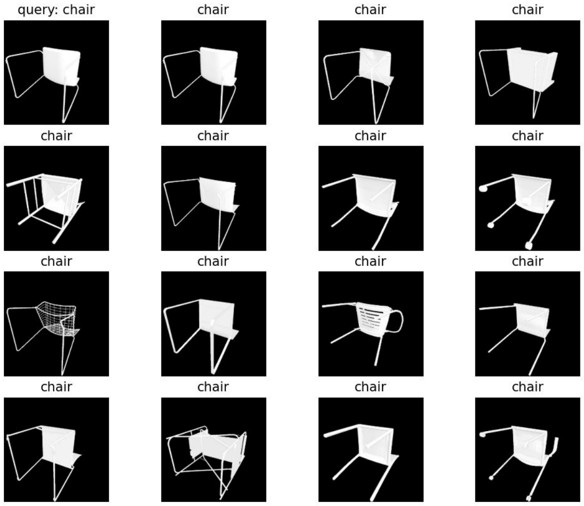


Fig. 5. Retrieval results when the query is a chair.

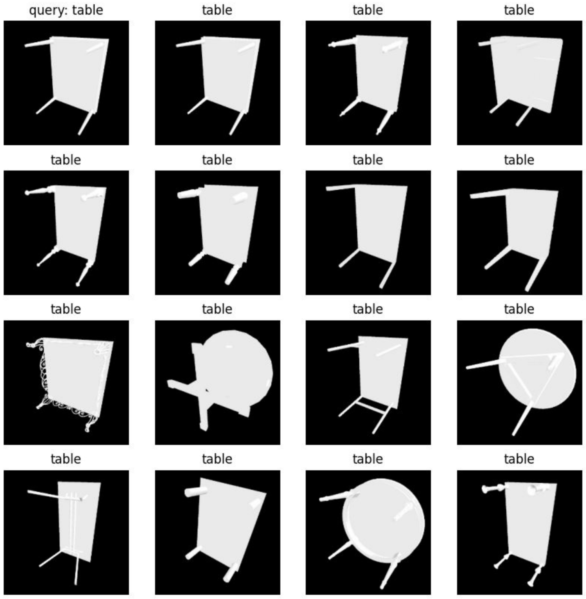


Fig. 6. Retrieval results when the query is a table.

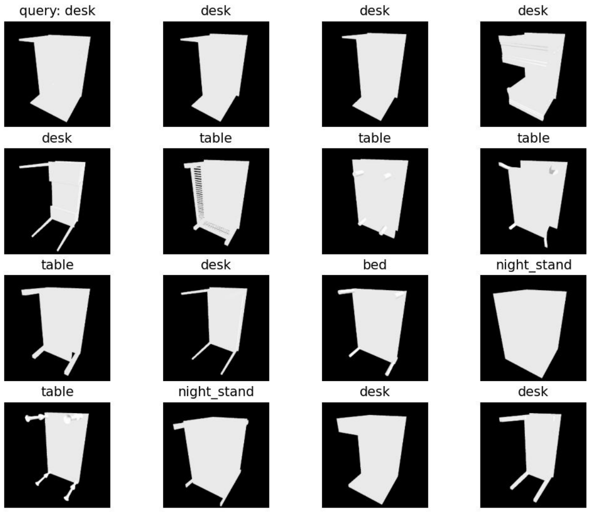


Fig. 7. Retrieval results when the query is a desk.

method is that it only has to train one CNN while the other multi-view CNN based methods must train several in parallel. This results in a lower computation time, training time and memory requirements. Some of the drawbacks of the proposed method is that its retrieval performance is not competative with the state of the art methods. One of the reasons for this is because it only trains the neural network using one image taken from one viewpoint while the other models use many images taken from different viewpoints. Since some objects look similar when they are viewed from the same angle it is difficult for the neural network to distinguish them if given only one viewpoint. One way to fix this issue is to train the neural network using many viewpoints at once and aggregate the output of the second to last fully connected layer from each CNN into a single discriptor. Another improvement that could be made is to use a state of the art CNN from the 2D image classification space, such as ResNet50, instead of the outdated AlexNet CNN that is being used.

References

[1] L. Hoang, S.-H. Lee, O.-H. Kwon, and K.-R. Kwon, "A deep learning method for 3D object classification using the wave kernel signature and a center point of the 3D-triangle mesh," in *Electronics,* vol. 8, no. 10, p. 1196, 2019.

[2] L. Hoang, S.-H. Lee, and K.-R. Kwon, "A 3D shape recognition method using hybrid deep learning network CNN–SVM," in *Electronics,* vol. 9, no. 4, p. 649, 2020.

[3] M. Mirbauer, M. Krabec, J. Krivanek, and E. Sikudova, "Survey and evaluation of neural 3D shape classification approaches," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* pp. 1–1, 2021.

[4] B. Ding, L. Tang, and Y.-jun He, "An efficient 3D model retrieval method based on Convolutional Neural Network," *Complexity,* vol. 2020, pp. 1–14, 2020.

[5] A. Angrish, A. Bharadwaj, and B. Starly, "MVCNN++: Computer-aided design model shape classification and retrieval using multi-view convolutional neural networks," *Journal of Computing and Information Science in Engineering,* vol. 21, no. 1, 2020.

[6] I. O. Taybi, T. Gadi and R. Alaoui, "2DSlicesNet: A 2D Slice-Based Convolutional Neural Network for 3D Object Retrieval and Classification," *IEEE Access,* vol. 9, pp. 24041-24049, 2021, doi: 10.1109/ACCESS.2021.3056613.

[7] Q. Chen and Y. Chen, "Multi-view 3D model retrieval based on enhanced detail features with Contrastive Center loss," *Multimedia Tools and Applications,* 2022.

[8] K. Sfikas, I. Pratikakis, and T. Theoharis, "Ensemble of panorama-based convolutional neural networks for 3D model classification and retrieval," *Computers & Graphics,* vol. 71, pp. 208–218, 2018.

[9] B. Ding, L. Tang, Z. Gao and Y. He, "3D Shape Classification Using a Single View," *IEEE Access,* vol. 8, pp. 200812-200822, 2020, doi: 10.1109/ACCESS.2020.3035583.

[10] F. Gomez-Donoso, F. Escalona, S. Orts-Escolano, A. Garcia-Garcia, J. Garcia-Rodriguez and M. Cazorla, "3DSliceLeNet: Recognizing 3D Objects Using a Slice-Representation," *IEEE Access,* vol. 10, pp. 15378-15392, 2022, doi: 10.1109/ACCESS.2022.3148387.

[11] S. Watanabe, S. Takahashi and L. Wang, "Aggregating Viewpoints for Effective View-Based 3D Model Retrieval," *2021 25th International Conference Information Visualisation (IV),* 2021, pp. 320-327, doi: 10.1109/IV53921.2021.00058.

[12] C. Ma, Y. Guo, J. Yang and W. An, "Learning Multi-View Representation With LSTM for 3-D Shape Recognition and Retrieval," *IEEE Transactions on Multimedia,* vol. 21, no. 5, pp. 1169-1182, May 2019, doi: 10.1109/TMM.2018.2875512.

[13] Y. Zhou, F. Zeng, J. Qian, and X. Han, "3D shape classification and retrieval based on Polar View," *Information Sciences,* vol. 474, pp. 205–220, 2019.