

**Figure 2.** Architectures of Neural Network based on Point Cloud and Image. Note that they have similar hyper-parameters and structures.

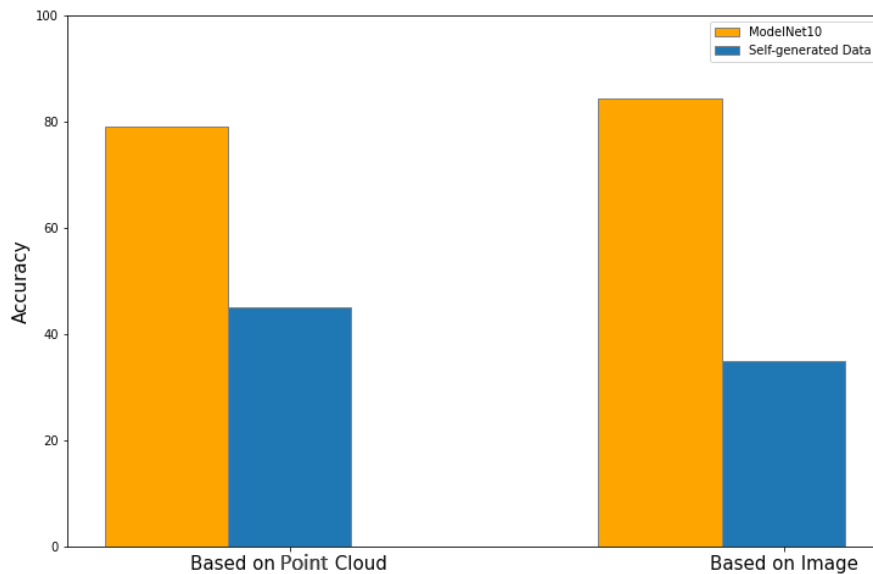
We used 2D convolutional neural network to classify the 2D images. The building blocks of a convolutional neural network are convolutional layers. 2D convolutional layers designed for classifying images extract features with filters. These filters are two-dimensional matrices that show features after applying them to the original pictures. After the convolutional layers, the filters are flattened and connected to fully connected layers to learn these features and output. We designed our 2D convolutional neural network with the same number of convolutional layers as the 3D neural network. The convolutional layers connect to a 2D max pooling layer, followed by the same fully connected layers at the end of the network.

We created neural networks very similar to each other to control the effect of hyper-parameters on performance (Figure 2). The only difference between the networks is that they used the different convolutional layers and used them for various purposes. For example, the 3D classification neural network used the 1D convolutional layers to extend the dimension of the points for max pooling, and the 2D convolutional neural network used 2D convolutional layers to extract features from the images.

## Results and Analysis

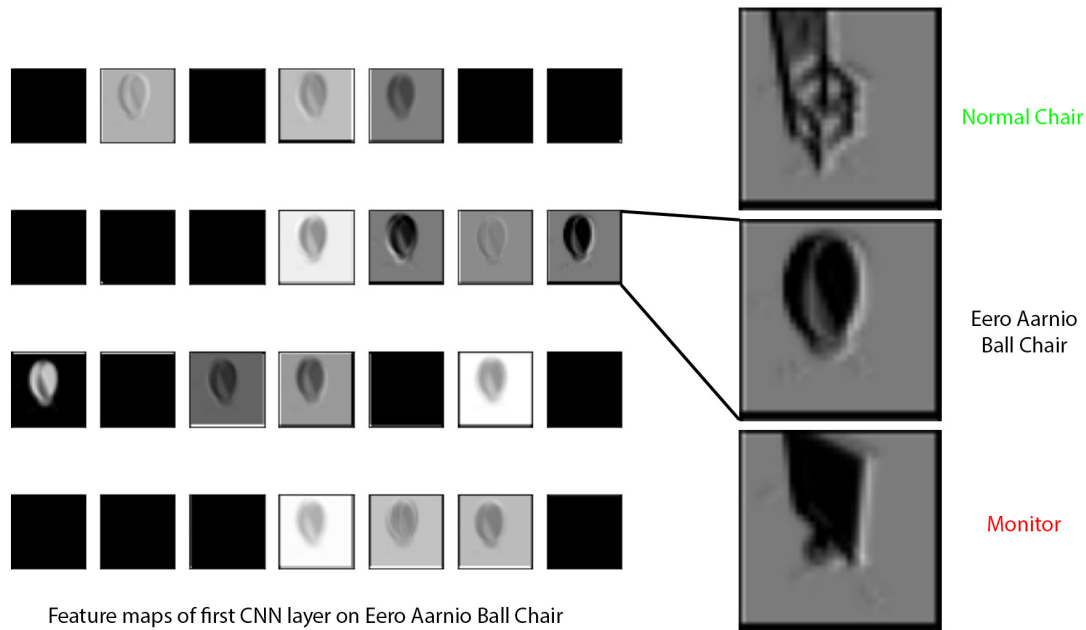
We observed that both point cloud based and image based neural networks performed well on the testing data included in the ModelNet10 dataset: 79.07% accuracy for classification based on point clouds and 81.5% accuracy for classification based on images. However, when the neural networks classified the generated data based on more abstracted designed furniture, the performance decreases by 35-45% accuracy in Figure 3. Furthermore, the performance of the 2D neural network is worse than the performance of the 3D neural network with a 10% gap on the new data, showing

that classification based on point cloud extracts more features that help classify these abstract design furniture compared to classification based on images.



**Figure 3.** Comparison the accuracy between classification based on point clouds and images for ModelNet10 and self-generated datasets.

So why are the neural networks failing to classify the furniture that we generated? We looked into the feature maps of the 2D convolutional neural network that we created for processing 2D images. Figure 4 shows that the first convolutional layer is already failing to extract correct features to classify the Eero Aarnio ball chair as a chair. The ball-shaped body does not resemble the appearance of a chair; however, the feature of the Eero Aarnio ball chair is closely related to the feature extracted from a monitor image. The chair’s foot is similar to a monitor stand. The 2D convolutional neural network gave us a prediction of 64.87% as a monitor. On the 3D end, we observed the same thing happening. The neural network processing 3D point cloud failed to distinguish this chair and gave it 76.57% as a monitor, an even higher percentage on the failed prediction.



**Figure 4.** Feature comparison for image-based classification on Eero Aarnio chair. Note that the correct label is colored in green and incorrect label in red.

We observed the training process of each neural network on the same data set. The accuracy trend on the testing set of ModelNet10 is very similar between using 3D data and 2D data. However, these two neural networks converge differently. For example, CNN can come to the high prediction on the pictures with a low learning rate  $\alpha = 0.00001$  in 20 epochs. On the other hand, the neural network based on the point cloud requires a higher learning rate  $\alpha = 0.001$  to achieve the same performance with the same number of epochs. This results in the accuracy of the testing set going up and down from overshooting when making gradient descent. In addition, it shows that neural networks based on 3D point clouds are more challenging to converge than 2D images.

Time-wise, the image-based neural network wins with the total training time on the same amount of data being 3 times faster than the neural network trained on the point cloud using the same hardware. The time it took to preprocess the data for training also favors the 2D algorithm, averaging half of time compared to 3D. However, it took us around 10 min to generate photos from the 3D model format. To create point clouds from the same format, it took around 1 min.

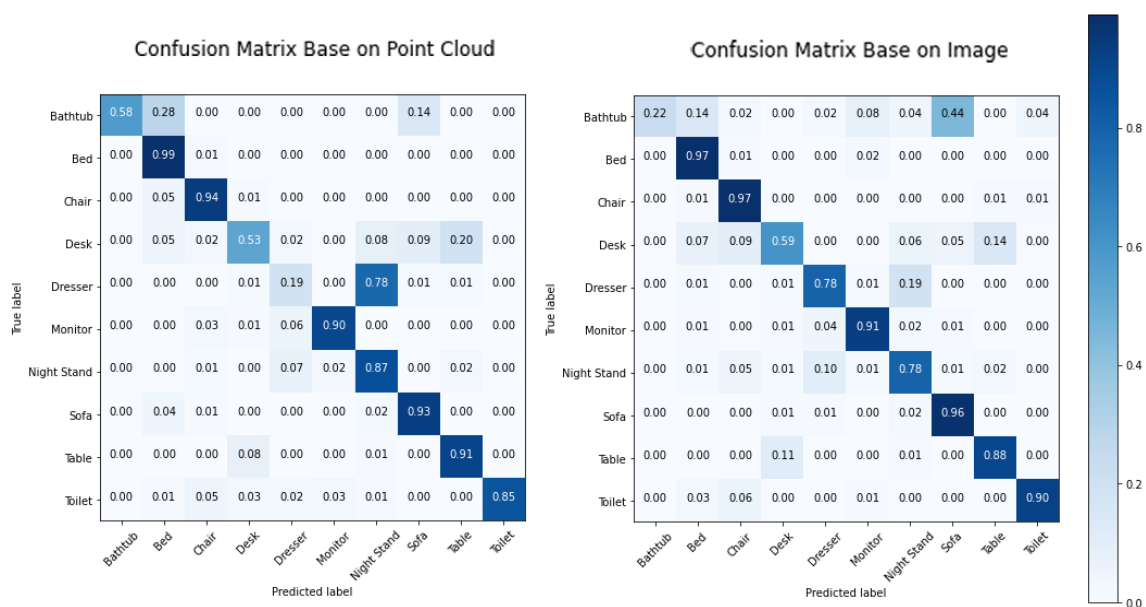


Figure 5. Confusion Matrices of classification based on point clouds and images.

For the performance of each different class in the testing set of ModelNet10, we found an interesting correlation between the results of 3D classification on point clouds and 2D classification on images in Figure 5. In addition, we observed that both classification algorithms have similar false positives and classes with the highest or lowest accuracy. This trend tells us that both neural networks are picking up similar features that contribute to their predictions. However, the classification based on the point cloud performs poorly in some classes compared to images. For example, the neural network based on the point cloud predicts most dressers as nightstands, which is due to us normalizing the point cloud and losing the scale of its original model. This does not happen to images because the image generated from the model still conserves its original size. In addition, normalizing the images scales down the RGB value instead of the coordinates for point clouds. We also observed that classification based on images is significantly worse at distinguishing between a bathtub and a sofa. The cause of this phenomenon is the fact that images do not contain any 3D information. Bathtubs are curved down on the top so they can hold water. Images won't be able to record all the curves in bathtubs that are blocked. Therefore, it is hard for the neural network to extract features corresponding to that curve with just an image from one direction.

## Discussion

We observed a significant performance drop when our algorithm performs on abstract objects. We believed that other than research around the performance of these algorithms with losing data, we should put more effort into researching the influence of different designs. Computer vision used in autonomous vehicles and robots needs to classify not only living objects but also artificial objects. Animals are generic enough for neural networks to classify them with minimal errors. However, artificial things, especially furniture, are a lot trickier than animals because they are influenced by human design. Design is not easy to predict, making it hard to extract features from them similar to an ordinary object. Nevertheless, we still believe that there are ways of classifying them using neural networks. Using videos instead of static data like images and point clouds, deep neural networks can analyze human interaction with the object to understand these abstract objects' functions and categories. However, further research is still needed in this area.

From the designer's perspective, we believe that the extensive application of artificial intelligence using computer vision will significantly influence designers' creativity. This is because designers have to consider not only human interaction with the object but also computer interactions. Considering that computers currently are not as knowledgeable as humans on how to interact with manufactured objects, it will be challenging to design something in a world with humans and computers. Therefore, we suspect that computer vision will limit the potential of abstract designs.

## Conclusion

In this project, we created our neural networks classifying two different data types, 3D point clouds and 2D images. We then trained the neural networks on the same 3D models from ModelNet10. For 3D point clouds, we used trimesh to convert the 3D models into point coordinates and feed them into a neural network based on PointNet Vanilla. And for 2D images, we rendered the models and took pictures from the same perspective and then input them into a 2D convolutional neural network. After training both neural networks, we tested their performance on the ModelNet10 testing set and the 3D models we generated. We generated these 3D models based on abstract designs from the categories of ModelNet10 and converted them into the same format as other testing data. To get the result, we compared the performance of the classification neural networks based on point cloud and images on ModelNet10 and the self-generated data.

We found that the classification based on point cloud and images are pretty similar performance-wise on the ModelNet10. However, when we input self-generated data, the point cloud based neural network has significantly higher accuracy than the image based neural network. Classification based on images also takes less time to train and is easier to converge than classification based on point clouds.

We conclude that though both point cloud and image based classification algorithms have similar performance, it seems more beneficial to combine these techniques for better performance and scalability. Furthermore, classifying abstract objects is very inefficient with both using point clouds and images. We suspect that we can minimize the performance lost on those objects with the help of deep neural networks and contextual analysis. However, further research is needed.

## Code

Github Repository: [github.com/M4rkX-Y/Point-Cloud-vs.-Image-Classification](https://github.com/M4rkX-Y/Point-Cloud-vs.-Image-Classification)

## Acknowledgments

I would like to express my sincere appreciation to Dr. Guillermo Goldsztein and TA Davida Kollmar, who gave me the golden opportunity to write this paper on machine learning. Secondly, I would also like to thank my family and friends, Sam Liu and James Li, for peer-reviewing the paper. Thanks again to all who supported me.

## References

Dawson-Haggerty et al. (n.d.). trimesh. Version 3.2.0. url: <https://trimsh.org/>

Feng, Yutong, Yifan Feng, Haoxuan You, Xibin Zhao, and Yue Gao (2018). MeshNet: Mesh Neural Network for 3D Shape Representation. arXiv: [1811.11424 \[cs.CV\]](https://arxiv.org/abs/1811.11424)



Filgueira, A., H. González-Jorge, S. Lagüela, L. Díaz-Vilariño, and P. Arias (2017). “Quantifying the influence of rain in LiDAR performance”. In: Measurement 95, pp. 143–148.

Qi, Charles R., Hao Su, Kaichun Mo, and Leonidas J. Guibas (2017). Point-Net: Deep Learning on Point Sets for 3D Classification and Segmentation. arXiv: [1612.00593 \[cs.CV\]](https://arxiv.org/abs/1612.00593)

Rusu, Radu Bogdan and Steve Cousins (2011). “3D is here: Point Cloud Library (PCL)”. In: 2011 IEEE International Conference on Robotics and Automation, pp. 1–4. doi: [10.1109/ICRA.2011.5980567](https://doi.org/10.1109/ICRA.2011.5980567)

Sedaghat, N., Zolfaghari, M., Amiri, E., & Brox, T. (2016). Orientation-boosted voxel nets for 3d object recognition. arXiv preprint arXiv: [1604.03351\[cs.CV\]](https://arxiv.org/abs/1604.03351)

Wandinger, Ulla (2005). Introduction to Lidar. Ed. by Claus Weitkamp. New York, NY: Springer New York, pp. 1–18. isbn: 978-0-387-25101-1. doi: 10.1007/0-387-25101-4\_1. url: [https://doi.org/10.1007/0-387-25101-4\\_1](https://doi.org/10.1007/0-387-25101-4_1)

Wu, Zhirong et al. (2015). 3D ShapeNets: A Deep Representation for Volu-metric Shapes. arXiv: [1406.5670 \[cs.CV\]](https://arxiv.org/abs/1406.5670)

Xiao, Tong, Tian Xia, Yi Yang, Chang Huang, and Xiaogang Wang (2015). “Learning From Massive Noisy Labeled Data for Image Classification”. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)

Zhang, Feihu, Daniel Clarke, and Alois Knoll (2014). “Vehicle detection based on LiDAR and camera fusion”. In: 17th International IEEE Conference on Intelligent Transportation Systems (ITSC), pp. 1620–1625. doi: [10.1109/ITSC.2014.6957925](https://doi.org/10.1109/ITSC.2014.6957925)