3D OBJECT DETECTION

A BTP Report by

Harish Mullagura (S20190010124)
Sashidhar Motte (S20190010123)
Charan Surya Sajja (S20190010156)

Mentor: Dr. Piyush Joshi

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

DATE: 6th December 2022

2nd Semester Report

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRICITY

CANDIDATE’S DECLARATION

I hereby certify that the work which is being presented in the BTP entitled **“3D OBJECT DETECTION”** in the partial fulfillment of the requirements for the award of the degree of B.Tech and submitted in the Indian Institute of Information Technology SriCity, is an authentic record of my own work carried out during the time period from January 2022 to December 2022 under the supervision of Dr. Piyush Joshi, Indian Institute of Information Technology SriCity, India.

The matter presented in this report has not been submitted by me for the award of any other degree of this or any other institute.

Signature of the student with date

M. Harish
06/12/2022

Sashidhar
6/12/22

Charan
6/12/22

(Harish Mullagura/Sashidhar Motte/Charan Surya Sajja)

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Signature of BTP Supervisor

(Dr. Piyush Joshi)

ABSTRACT

3D Object Detection is a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos. Many object detection methods have been developed based on different types of data including image, radar, and lidar. Some recent works use point clouds for 3D object detection. In this report, we design a novel type of neural network that directly consumes point clouds, which well respects the permutation invariance of points in the input. 3D scene modeling has long been a fundamental problem in computer graphics and computer vision.

With the popularity of consumer-level RGB-D cameras, there is a growing interest in digitizing real-world indoor 3D scenes. However, modeling indoor 3D scenes remains a challenging problem because of the complex structure of interior objects and poor quality of RGB-D data acquired by consumer-level sensors. Various methods have been proposed to tackle these challenges. Our network, named PointNet, provides a unified architecture for applications ranging from object classification, part segmentation to scene semantic parsing. PointNet is highly efficient and effective in real-world indoor 3D scenes.

Contents

1. Introduction

Consumer-level color and depth (RGB-D) cameras (e.g., Microsoft Kinect) are now widely available and are affordable to the general public. Ordinary people can now easily obtain 3D data from their real-world homes and offices. Meanwhile, other booming 3D technologies in areas such as augmented reality, stereoscopic movies, and 3D printing is also becoming closer to our daily life. We are living on a “digital Earth”. Therefore, there is an ever-increasing need for ordinary people to digitize their living environments.

In contrast, while the growth of 3D digital models has accelerated over the past few years, the growth remains comparatively slow, mainly because making 3D models is a demanding job which requires expertise and is time consuming. Fortunately, the availability of low cost RGB-D cameras along with recent advances in modeling techniques offers a great opportunity to change this situation. In the longer term, 3D big data has the potential to change the landscape of 3D visual data processing.

With the advent of metaverse, virtual reality has become very popular. It has the potential to change people's perception of technology. The vision of the metaverse is very far from reality right now. Building such a big thing is a tedious and time taking task. For example, if we want to build a virtual world around IIIT Sri City it would require someone to replicate the 3D model of the whole environment (campus).

This can be time consuming and is a very hectic task. The problem is not just with metaverse it’s about any VR based technology or building any open-world games.

What if we can build those virtual environments or 3D models in a very little span of time?

The metaverse requires creation of complex environments with 3D modeling which is a tedious process. However, the 3D scene recreation can help in building these complex environments in no time.

To build a model, when the input is given as RGB-D image or a real time video it should be able to convert the scenario or frame into a 3D model which can further be used for various applications in virtual reality or augmented reality based technologies.

With this project we would like to create such a 3D scene recreation model where the outcome can be used in the metaverse or virtual reality based applications.

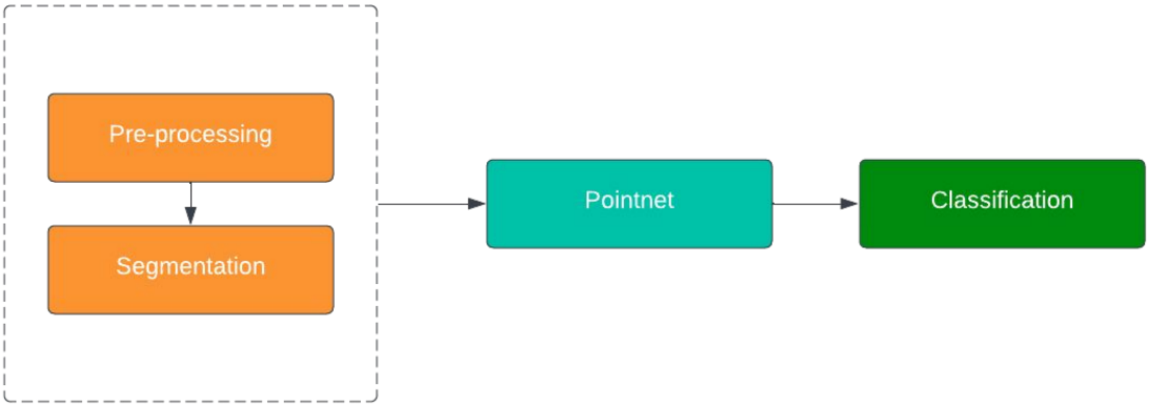


Figure 1: Overview of the project

2. Literature Survey

Author	Year	Title	Dataset	Method	Drawbacks
Kang Chen, Yu-Kun Lai, Shi-Min Hu	2015	3D indoor scene modeling from RGB-D data	ModelNet 40	Survey of existing methods	Large scale Indoor scenes cannot be classified.
Bin Yang, Wenjie Luo, Raquel Urtasun	2018	PIXOR: Real-time 3D Object Detection from Point Clouds	KITTI Raw	PIXOR Accuracy :- 81.38	Will fail when there are no observed Lidar points. In longer ranges, object localization becomes inaccurate.
Shishun Zhang , Longyu Zheng, Wenbing Tao	2021	Survey and Evaluation of RGB-D SLAM	TUM RGB-D	RGB-D Slam Accuracy :- 85.6	Invariable gray level is a strong assumption that is difficult to satisfy.
Daniel Maturana,Sebastian Scherer	2015	VoxNet: A 3D Convolutional Neural Network for Real-Time Object Recognition	ModelNet 10 ModelNet 40	VoxNet Accuracy:- 83%	Conversion to voxels requires higher computation power for larger datasets.
Charles R. Qi, Hao Su, Kaichun Mo, Leonidas J. Guibas	2017	PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation	ModelNet 40	Pointnet Accuracy :- 89.2	It does not consider local structure induced by the metric space made by its local neighbors. Experiments exhibit PointNet is expressively sensitive to the hyper-parameters.

Table 1: Literature Survey

2.1 PointNet

PointNet is a unified architecture that directly takes point clouds as input and outputs either class labels for the entire input or per point segment or part labels for each point of the input. The basic architecture of our network is surprisingly simple as in the initial stages each point is processed identically and independently. In the basic setting each point is represented by just its three coordinates (x, y, z). Additional dimensions may be added by computing normals and other local or global features. This approach is the use of a single symmetric function, max pooling. Effectively the network learns a set of optimization functions or criteria that select interesting or informative points of the point cloud and encode the reason for their selection. The final fully connected layers of the network aggregate these learnt optimal values into the global descriptor for the entire shape as mentioned above are used to predict per point labels. We propose a novel deep net architecture that consumes raw point cloud (set of points) without voxelization or rendering. It is a unified architecture that learns both global and local point features, providing a simple, efficient and effective approach for a number of 3D recognition tasks.

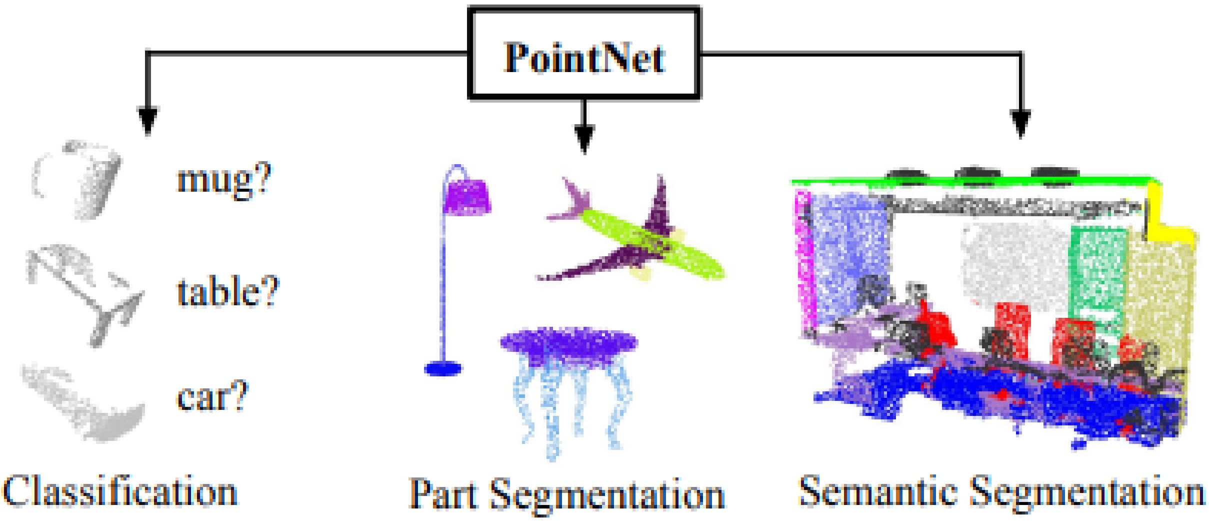


Figure 2: Application of PointNet

2.2 Part Segmentation

Part segmentation is a challenging fine-grained 3D recognition task. Given a 3D scan or a mesh model, the task is to assign part category labels to each point or face. We formulate part segmentation as a per point classification problem. Evaluation metric is mIoU on points. For each shape S of category C, to calculate the shape's mIoU: For each part type in category C, compute IoU between ground truth and prediction. If the union of ground truth and prediction points is empty, then count part IoU as 1. Then we average IoUs for all part types in category C to get mIoU for that shape. To calculate mIoU for the category, we take the average of mIoUs for all shapes in that category.

2.3 Semantic Segmentation

Our network on part segmentation can be easily extended to semantic scene segmentation, where point labels become semantic object classes instead of object part labels. To prepare training data, we firstly split points by room, and then sample rooms into blocks with area 1m by 1m. We train our segmentation version of PointNet to predict per point class in each block. Based on the semantic segmentation output from our network, we further build a 3D object detection system using connected components for object proposal.

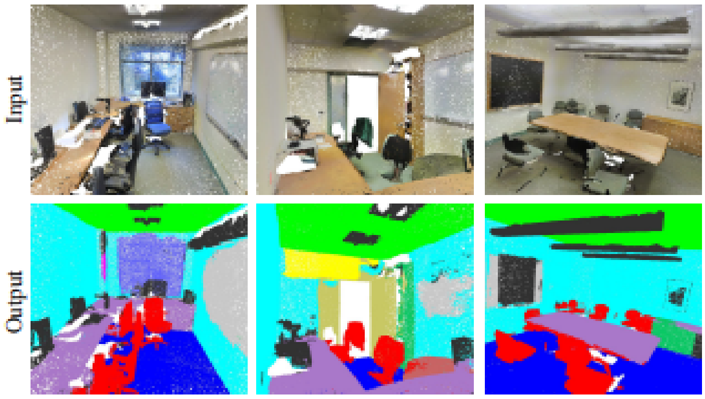


Figure 3: Semantic Segmentation

2.4 ModelNet10 Dataset

We have used a dataset called ModelNet10. ModelNet10 dataset is a part of ModelNet40 dataset, containing 4,899 pre-aligned shapes from 10 categories. There are 3,991 (80%) shapes for training and 908 (20%) shapes for testing. The CAD models are in Object File Format (OFF). To build the core of the dataset, a list of the most common object categories in the world was compiled, using the statistics obtained from the SUN database.

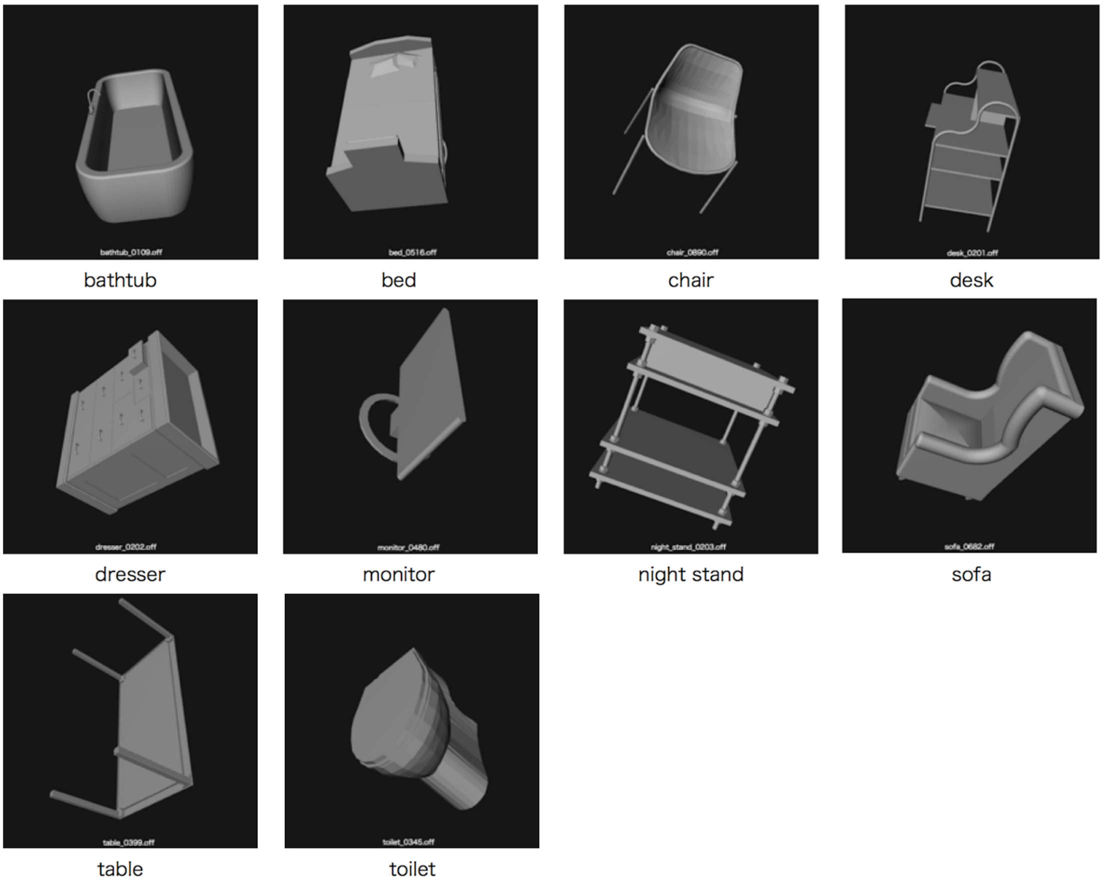


Figure 4: ModelNet 10 Dataset

3. Methodology

The method we have used in the application is PointNet. The motivation behind this application is to classify and segment 3D representation of images. They use a data structure called Point cloud, which is a set of points that represents a 3D shape or an object.

Many other techniques convert the point cloud into some other representation called voxel (volumetric pixel) before it is fed into the Deep neural networks. However, such transformation leads data too voluminous, and introducing quantization to the 3D structure can also lead to variance from natural artifacts.

In this project,we used PointNet for directly consuming Point clouds and output the relevant classification of image or segmentation. We have taken Point Sets from Point cloud as input. The point cloud is represented by a set of 3D points Pi where each point is represented as(xi, yi, zi).

For the object classification task, the input point cloud is directly sampled from the shape or pre-segmented from the scene point cloud. Properties of Point Sets are permutation Invariance, Transformation Invariance and Interaction between different points. For semantic segmentation, the input can be a single object from the part region segmentation or a small part of the 3D scene from the Object region segmentation.

The PointNet architecture is quite intuitive. The classification network uses a shared multi-layer perceptron to map each of the n points from 3 dimensions to 64-dimension. It's important that a single multi-layer perceptron is shared for each of the n points. Similarly, in the next layer, each n point is mapped from 64 dimensions to 1024 dimensions.

Now, we apply max-pooling to create a global feature vector in $\mathbb{R}^{1024}$. Finally, a three-layer fully-connected network (FCNs) is used to map the global feature vector to k output classification scores.

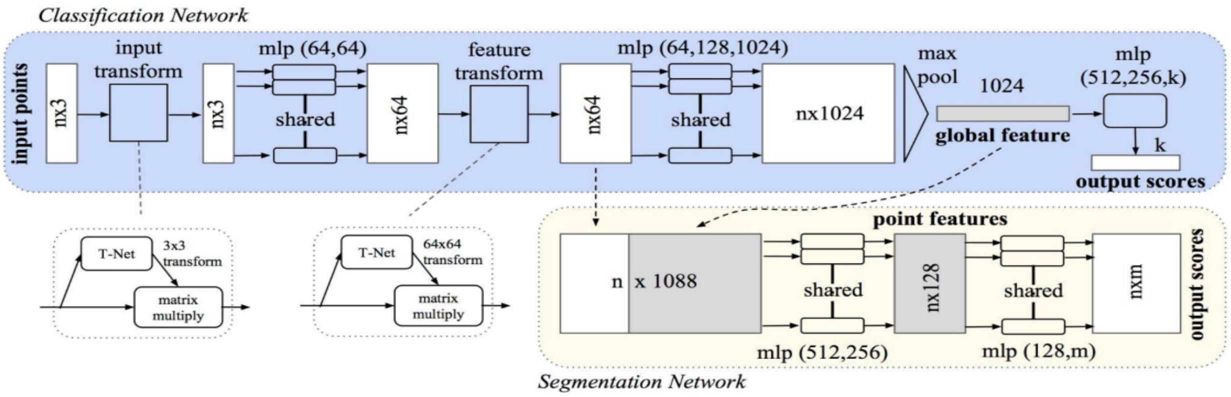


Figure 5: PointNet Architecture

The PointNet architecture has these key modules: the max-pooling layer, a local and global combination structure, and two joint alignment networks that align both local and global networks.

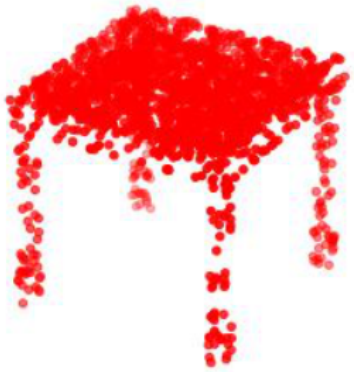


Figure 6: Point cloud

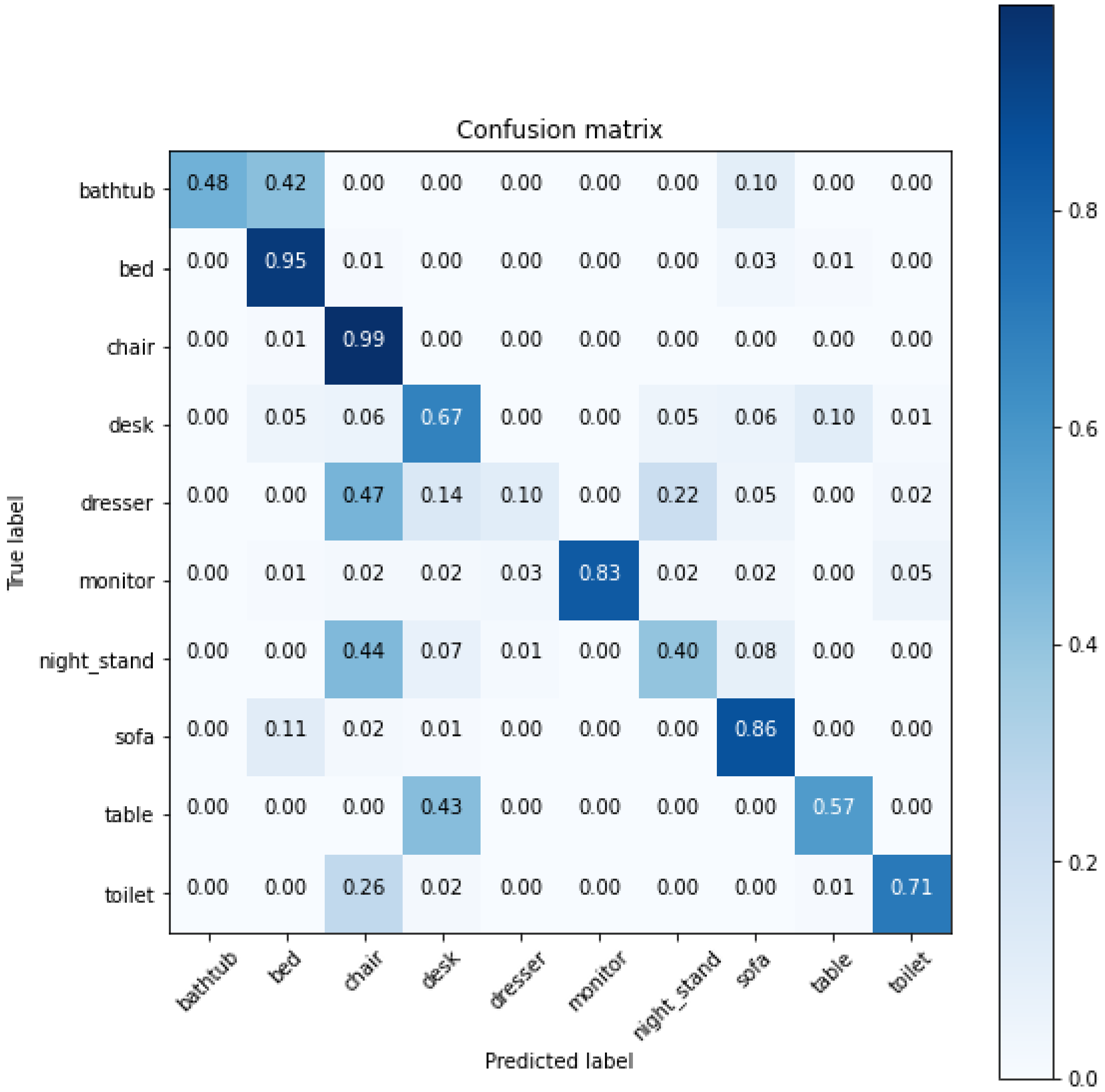
Finally it will classify the given object and the classified Object which is in .off format will be converted to .obj format using the ModelParser.

4. Results

We have trained a model that classifies the Object by using the Dataset ModelNet 10. We have used 4 epochs where the accuracy follows:

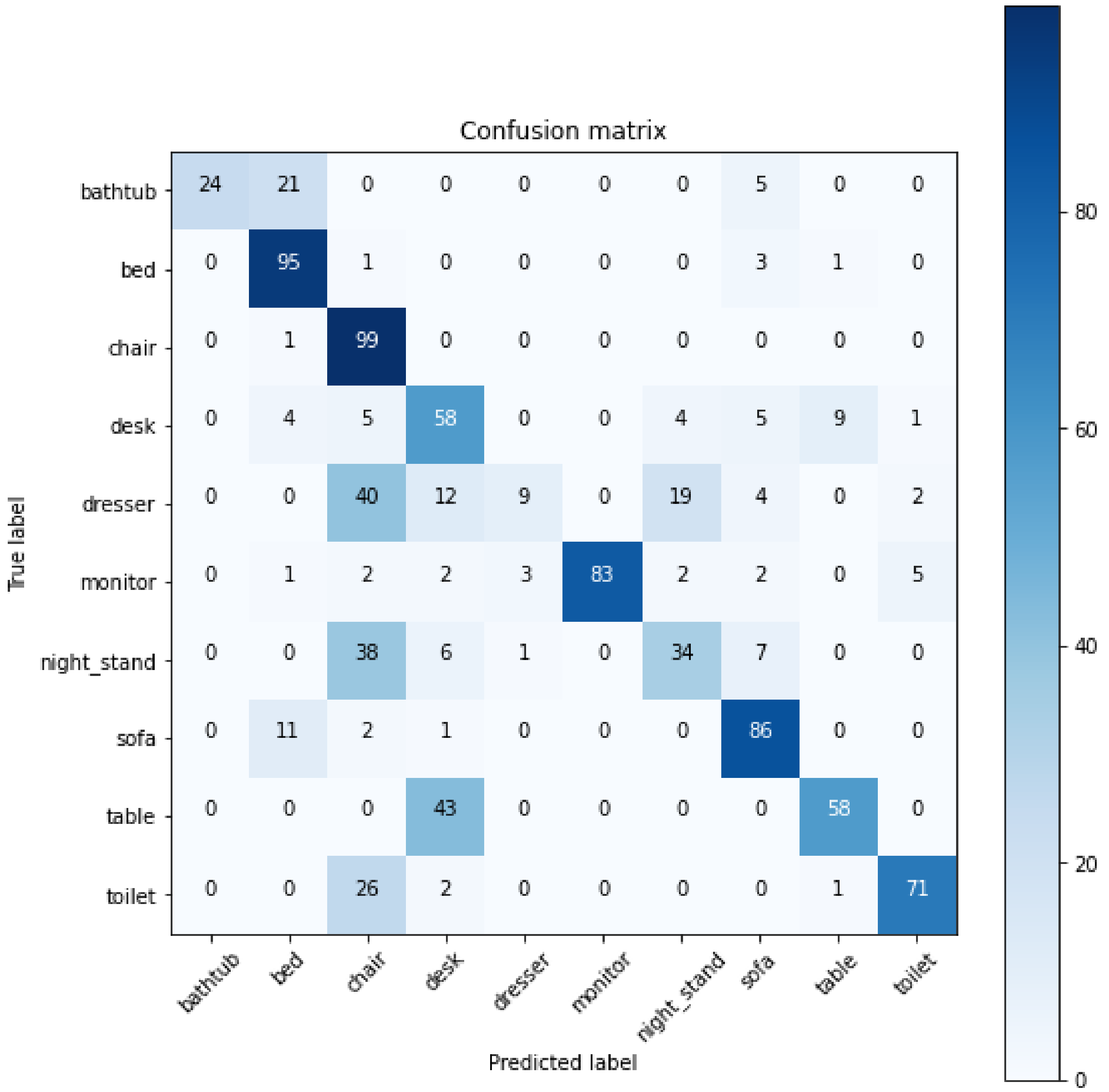
Epoch 1: 63%
Epoch 2: 70%
Epoch 3: 77%
Epoch 4: 80%

The PointNet Model got an accuracy of 80%.



	bathtub	bed	chair	desk	dresser	monitor	night_stand	sofa	table	toilet
bathtub	0.48	0.42	0.00	0.00	0.00	0.00	0.00	0.10	0.00	0.00
bed	0.00	0.95	0.01	0.00	0.00	0.00	0.00	0.03	0.01	0.00
chair	0.00	0.01	0.99	0.00	0.00	0.00	0.00	0.00	0.00	0.00
desk	0.00	0.05	0.06	0.67	0.00	0.00	0.05	0.06	0.10	0.01
dresser	0.00	0.00	0.47	0.14	0.10	0.00	0.22	0.05	0.00	0.02
monitor	0.00	0.01	0.02	0.02	0.03	0.83	0.02	0.02	0.00	0.05
night_stand	0.00	0.00	0.44	0.07	0.01	0.00	0.40	0.08	0.00	0.00
sofa	0.00	0.11	0.02	0.01	0.00	0.00	0.00	0.86	0.00	0.00
table	0.00	0.00	0.00	0.43	0.00	0.00	0.00	0.00	0.57	0.00
toilet	0.00	0.00	0.26	0.02	0.00	0.00	0.00	0.00	0.01	0.71

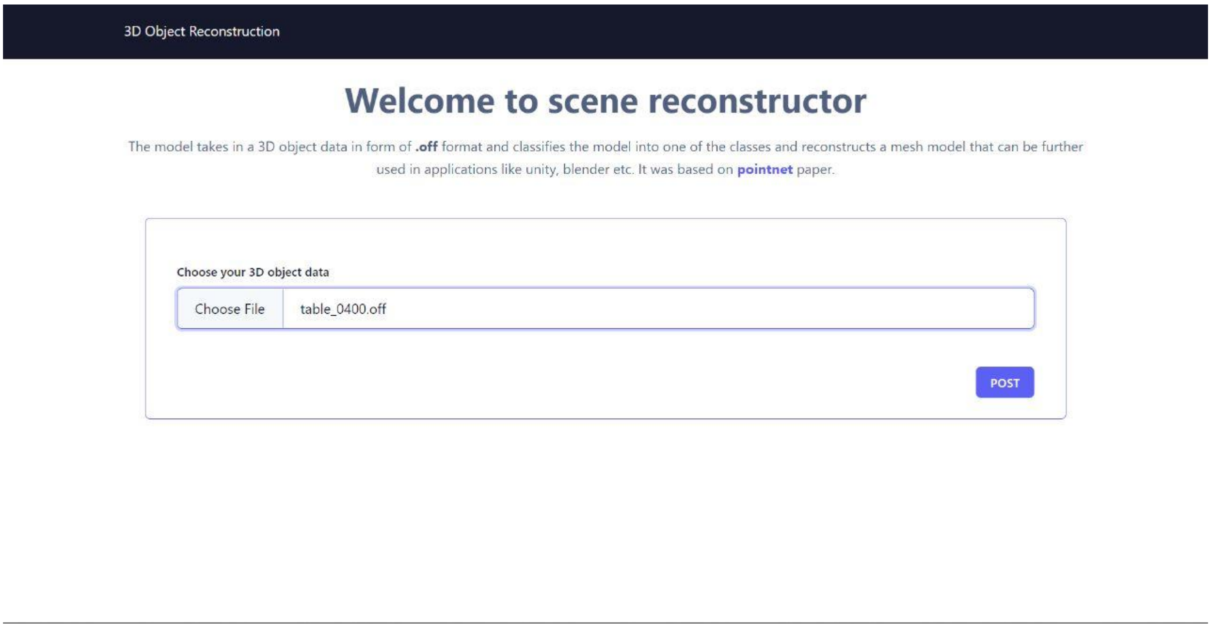
Figure 7: Normalized Confusion Matrix



	bathtub	bed	chair	desk	dresser	monitor	night_stand	sofa	table	toilet
bathtub	24	21	0	0	0	0	0	5	0	0
bed	0	95	1	0	0	0	0	3	1	0
chair	0	1	99	0	0	0	0	0	0	0
desk	0	4	5	58	0	0	4	5	9	1
dresser	0	0	40	12	9	0	19	4	0	2
monitor	0	1	2	2	3	83	2	2	0	5
night_stand	0	0	38	6	1	0	34	7	0	0
sofa	0	11	2	1	0	0	0	86	0	0
table	0	0	0	43	0	0	0	0	58	0
toilet	0	0	26	2	0	0	0	0	1	71

Figure 8: Confusion Matrix without Normalization

We have built an application where the input takes the .off (Object File Format) file and when we click on the post button, it classifies the object and will give the link of the converted mesh model to download. The downloaded file will be .obj(3D Object) file. It can be viewed in a 3D viewer. The downloaded .obj can now be used in applications like Unity.



3D Object Reconstruction

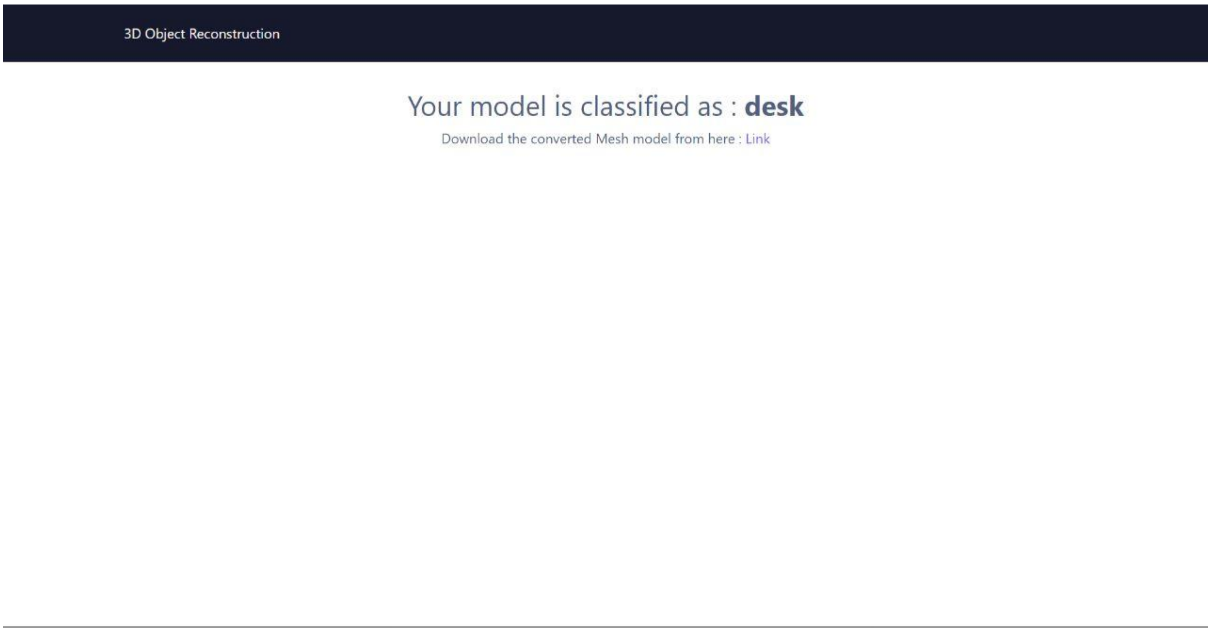
Welcome to scene reconstructor

The model takes in a 3D object data in form of **.off** format and classifies the model into one of the classes and reconstructs a mesh model that can be further used in applications like unity, blender etc. It was based on **pointnet** paper.

Choose your 3D object data

table_0400.off

Figure 9: Home page of web application

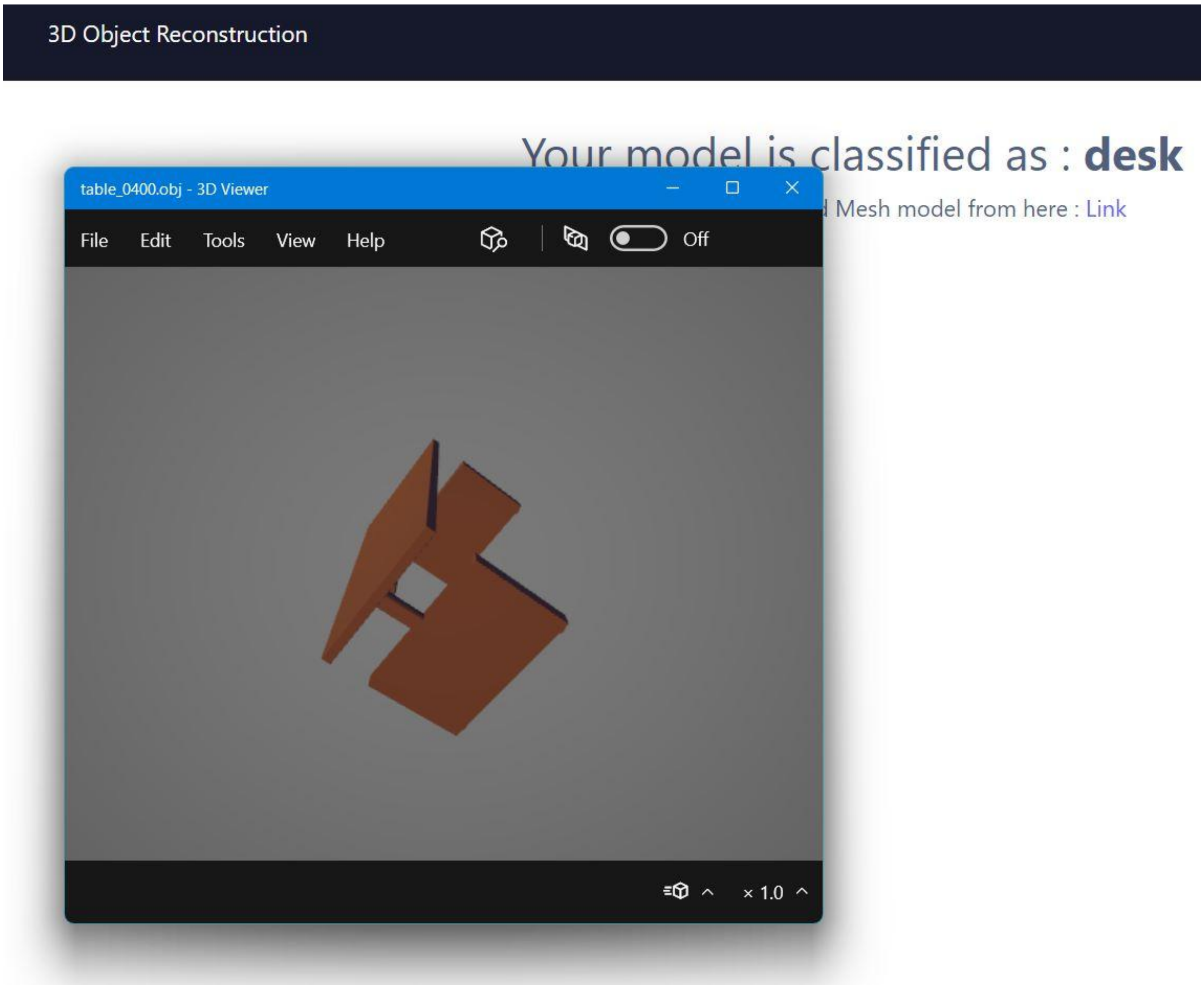


3D Object Reconstruction

Your model is classified as : **desk**

Download the converted Mesh model from here : Link

Figure 10: Classification of the model



3D Object Reconstruction

Your model is classified as : **desk**

Mesh model from here : Link

Figure 11: 3D View of the classified object

The above 3D object can be used in various applications in virtual reality or augmented reality based technologies.

Accuracy:-

Bathtub	: 93.4%
Bed	: 92.0%
Chair	: 97.2%
Desk	: 81.5%
Dresser	: 71.0%
Monitor	: 89.4%
Night Stand	: 56.0%
Sofa	: 86.9%
Table	: 93.4%
Toilet	: 95.9%

5. Conclusion

In the project, we have trained the model using the ModelNet 10 dataset with the help of the PointNet method. It is a deep neural network that can directly process point-clouds without using intermediate representations like voxels and it is suitable for object classification. The trained model will take the input as .off file and will classify the object and give the output as an option to download the .obj file which further can be used in applications like Unity for virtual reality and augmented reality. Further segmentation can be added for whole scene reconstruction by giving input as RGB-D image or a real time video where it should be able to convert the scenario or frame into a 3D model which can further be used for various applications in virtual reality or augmented reality based technologies.

6. Workflow

First Evaluation :- Literature survey and Concept learning.

Second Evaluation :- Explored datasets and code implementations of various papers.

Third Evaluation :- Finalized the pointnet paper and worked on the implementation of the model.

Fourth Evaluation :-

- □ Implemented the PointNet model using Pytorch
- □ Trained the Model and tested it on ModelNet10 dataset.
- □ Tweaked the model according to our application requirement.
- □ Built a web application.

7. Contributions

Name	Roll Number	Contribution
Harish Mullagura	S20190010124	● Explored different papers and implementations. ● Studied the **PIXOR** paper and implemented it using Pytorch. ● Trained and Tested the PIXOR model on KITTI dataset. ● Worked on the web application (Backend).
Sashidhar Motte Motte	S20190010123	● Explored different papers and implementations. ● Studied the **PointNet** paper and implemented it using Pytorch. ● Trained and Tested the PointNet model on ModelNet10 dataset. ● Worked on the web application (Backend + Model-Integration).
Charan Surya Sajja	S20190010156	● Explored different datasets and 3D models. ● Studied the **Voxnet** paper and implemented it using Tensorflow. ● Trained and Tested the Voxnet model on ImageNet dataset. ● Worked on the web application (Frontend).

8. List Of Figures

9. List Of Tables

10. List Of Abbreviations

3D	Three dimensional
RGB-D	Red Green Blue-Depth
VR	Virtual Reality
AR	Augmented Reality
MIOU	Mean Intersection Over Union
IOU	Intersection Over Union
CAD	Computer-Aided Design
OFF	Object File Format
OBJ	3D Object File
VOXEL	Volumetric Pixel
CNN	Convolutional Neural Network
FCN	Fully Convolutional Network

11. References

[1] Chen, K., Lai, Y. K., & Hu, S. M. (2015). 3D indoor scene modeling from RGB-D data: a survey. Computational Visual Media, 1(4), 267-278.
https://link.springer.com/article/10.1007/s41095-015-0029-x

[2] Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). Pointnet: Deep learning on point sets for 3d classification and segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 652-660).
https://arxiv.org/abs/1612.00593

[3] Bin Yang, Wenjie Luo, Raquel Urtasun; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 7652-7660
https://openaccess.thecvf.com/content_cvpr_2018/html/Yang_PIXOR_Real-Time_3D_CVPR_2018_paper.html

[4] S. Zhang, L. Zheng and W. Tao, "Survey and Evaluation of RGB-D SLAM," in IEEE Access, vol. 9, pp. 21367-21387, 2021, doi: 10.1109/ACCESS.2021.3053188.
https://ieeexplore.ieee.org/abstract/document/9330596

[5] Maturana, D., & Scherer, S. (2015, September). Voxnet: A 3d convolutional neural network for real-time object recognition. In 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS) (pp. 922-928). IEEE.
https://ieeexplore.ieee.org/abstract/document/7353481

[6] Open3d Documentation