# Improving LiDAR Point Cloud Classification of Urban Objects

Peggy (Yuchun) Wang
wangyuc@stanford.edu

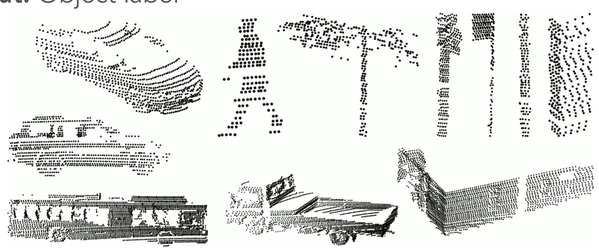
Tristan Gosakti
tgosakti@stanford.edu

## Introduction

- **Motivation**: LiDAR (Light Detection and Ranging) point cloud classification is important for autonomous driving since it helps with vehicle localization and obstacle avoidance
- **Baseline approach:** VoxNet - converts point clouds to 32x32x32 voxels, then use 3D CNNs to output object label
- **New approach:** Preprocessing - remove outliers from point cloud, model architecture changes
- **Results:** Removing Outliers Improves Accuracy

## Dataset

- **Sydney Urban Objects Dataset** = 631 Velodyne LiDAR point cloud objects
- **14** top object categories chosen - total of 588 point clouds
- Data Augmentation - each object was uniformly randomly translated and rotated for 12 times per point
- **5525/1957 = training/test split**
- x, y, z position for every point (N x 3 features)
- **Input:** N x 3 arrays -> 32 x 32 x 32 Voxel Occupancy Grid
- **Output:** Object label

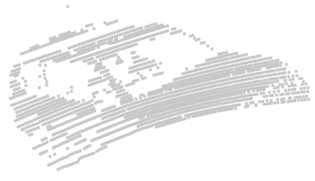


## Approach

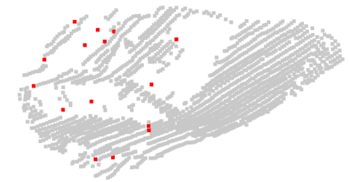
- **Remove outliers from point clouds before data augmentation**
  - removes points that are further away from their **3 nearest neighbors** compared to the average +/- **3.0 standard deviation** for the point cloud
- Data Augmentation: 1 point cloud -> 12 additional point clouds + original point cloud = 13 point clouds
- Voxelize Points
- Convert to 32 x 32 x 32 Occupancy Grid
- VoxNet

## Remove Outliers

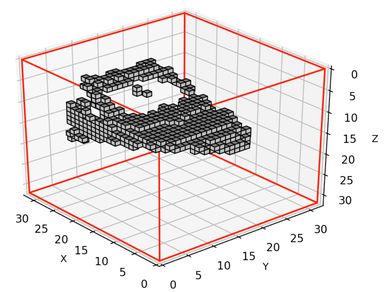
- Model: **VoxNet (baseline)**
- **Example for 4WD Object**



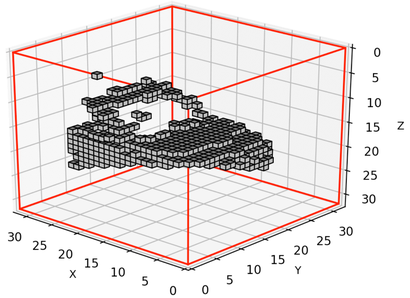
**Raw Point Cloud**



**Remove Outliers**



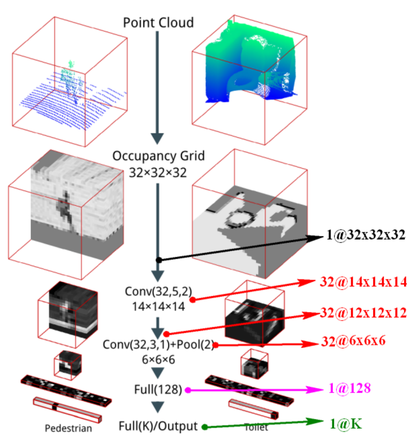
**Voxelization**



**Voxel without Removing Outliers**

## VoxNet Model and Hyperparameters

| Name | Values |
|---|---|
| Total Steps | 1379 |
| Epochs | 8 |
| Batch Size | 32 |
| Voxel Size | 24x24x24 |
| Voxel Zero Padding Size | 32x32x32 |
| Optimizer | Adam |
| Learning Rate | 0.001 |
| CNN Activation Function | Leaky RELU |
| FC Activation Function | RELU |
| Loss Type | Cross-entropy |




## Experiment Models

- Models: **VoxNet-DMP, VoxNet-Conv**
- Did not remove outliers
- Trained for **8** epochs each, batch size of **32**

| Name | VoxNet | VoxNet-DMP | VoxNet-Conv |
|---|---|---|---|
| Layer 1 | 3D Conv - 32F-5K-S2 | 3D Conv - 32F-5K-S2 | 3D Conv - 32F-5K-S2 |
| Layer 2 | 3D Conv - 32F-3K-S1 | Max Pool 2x2 - S2 | 3D Conv - 32F-3K-S1 |
| Layer 3 | Max Pool 2x2 - S2 | 3D Conv - 32F-3K-S1 | Max Pool 2x2 - S2 |
| Layer 4 | FC-128 | Max Pool 2x2 - S2 | 3D Conv - 32F-3K-S1 |
| Layer 5 | FC-14 | FC-128 | Max Pool 2x2 - S2 |
| Layer 6 | - | FC-14 | FC-128 |
| Layer 7 | - | - | FC-14 |

## Results

| Name | Training Acc | Testing Acc | Final Test Loss |
|---|---|---|---|
| VoxNet | 0.9231 | 0.6454 | 2.7060 |
| **VoxNet-Outliers** | **0.9750** | **0.6600** | 3.3907 |
| VoxNet-DMP | 0.9563 | 0.6510 | 2.0662 |
| VoxNet-Conv | 0.8995 | 0.6429 | 1.9999 |

## Conclusion and Future Work

- **Removing outliers improved VoxNet by 1.5%**
- Top accuracy is **66%**
- Simple Architecture changes and more layers do not make a big difference - perhaps because point cloud is sparse
- Future work includes:
  - Hyperparameter tuning, Regularization and Dropout
  - Different Model + Remove Outliers
  - Different network architectures, e.g. PointNet - train directly on point cloud
  - Add additional datasets - ShapeNet, Waymo Open Dataset