TOOPLOOX AI

Deep Learning for 3D World

Marcin Mosiołek

19/11/2018 PyData Warsaw

Intro

Marcin Mosiołek

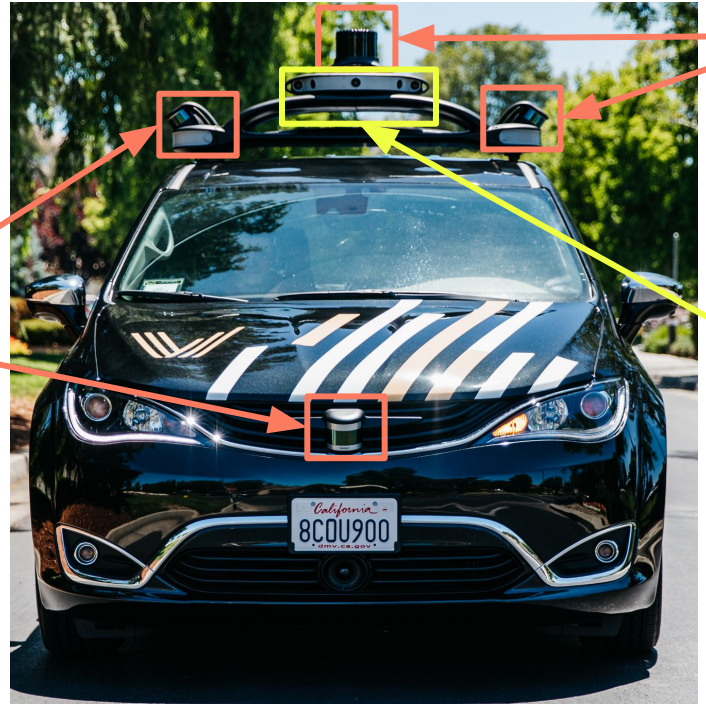
Machine Learning/AI Researcher @ **Tooploox**

Tooploox has been cooperating with Voyage: a Silicon Valley based company, whose mission is to super-charge communities with autonomous vehicles. We help them mainly with perception systems

Autonomous car’s perception system

Perception system includes both hardware and software and needs to spot and identify all the objects, which may affect driving decisions. In terms of machine learning it might be understood as object detection problem.

Autonomous car's perception system



Why there are so many LIDARs???

Autonomous car’s perception system

The main sensor is **LIDAR** (Light Detection and Ranging)

Advantages:

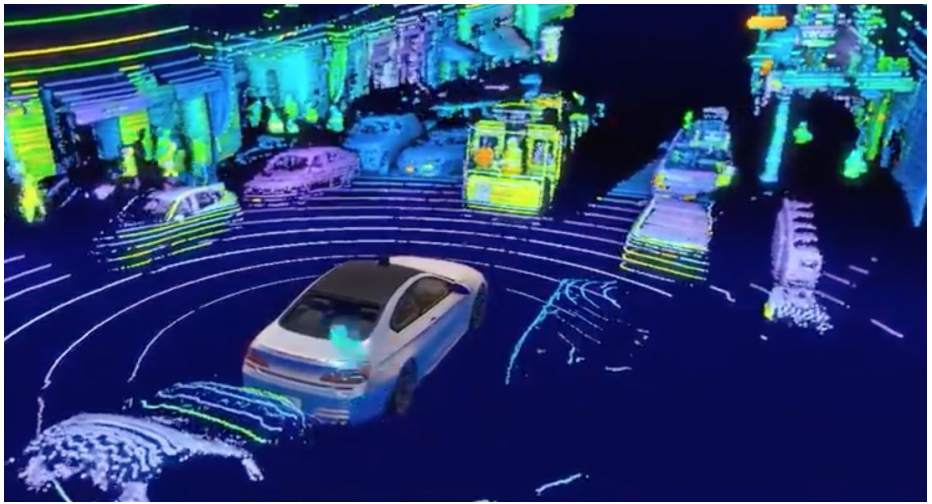
+ very accurate 3D map
+ light conditions independent
+ way higher resolution than RADAR
+ very long range, up to 300m

Disadvantages:

- color blind
- may have difficulties in heavy rain, or snow

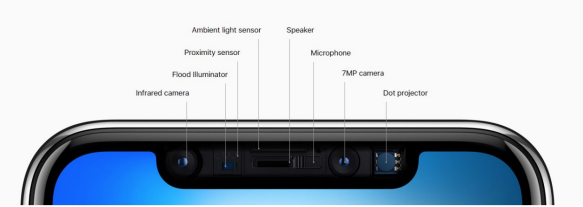
Point Cloud ~ raw sensor data

A set of 3D points where each point might be described by its coordinates plus extra feature channels such as color, reflectance, etc. Typically, there are around 100k points in a single scan.



There is plenty of other sensors...

RGB-Depth Cameras



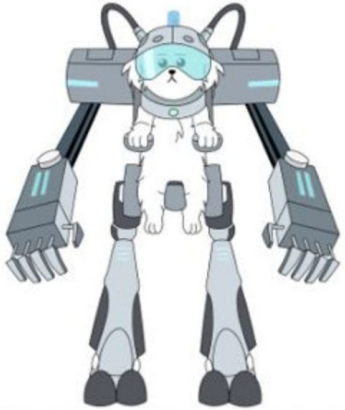
Mobile phones

Kinect

RealSense

Why do we need those sensors and algorithms?

Why 3D?



Robotics

Self driving

Augmented reality

Point clouds challenges

+ **Unordered and unstructured**: most of deep learning methods, focus on regular input representation

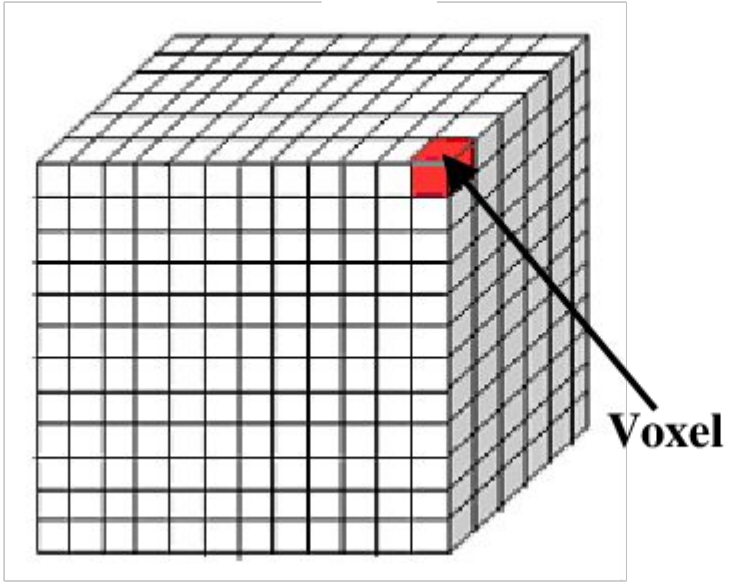
+ **Sparse and highly variable point density**: due to non-uniform sampling of 3D space, effective range of sensors, occlusion, relative pose.

+ **Interactions between points** needs to be captured

How to represent point clouds for ML?

Voxelization

A process which starts by dividing 3D space into a regular 3D grid. Cells of such grid are called Voxels



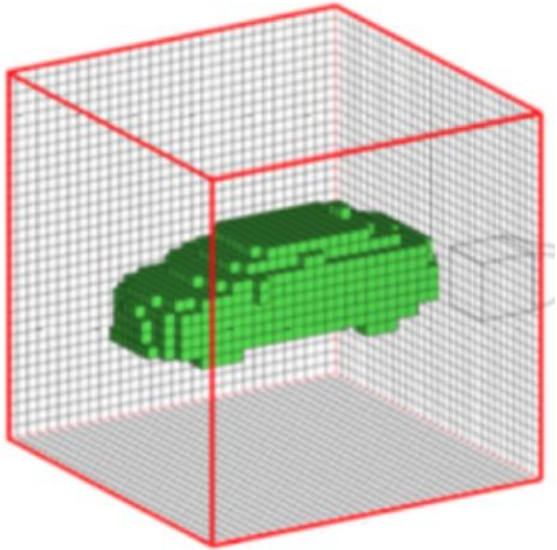
We may think of voxels as of pixels but in 3D

Voxels

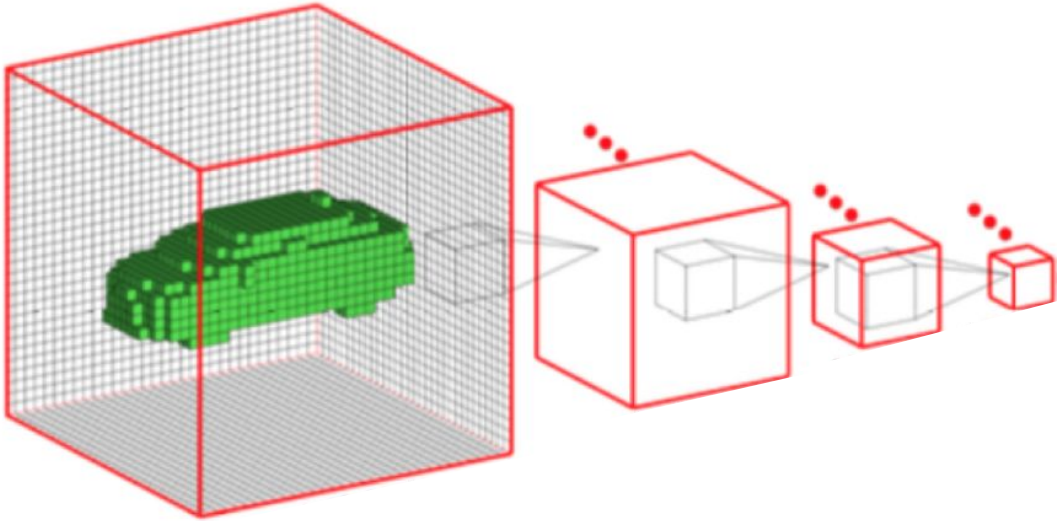
Point clouds elements are assigned to the corresponding voxels.

For each voxel there is calculated a feature vector.

In the simplest scenario, feature vector might be just one element: 0 or 1 indicating if there are points inside this voxel.



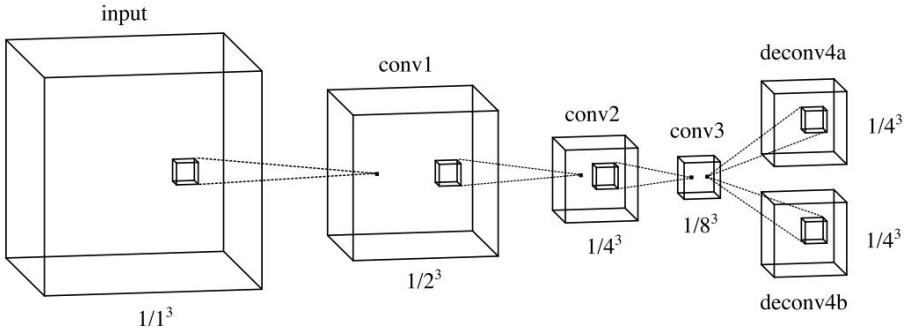
Machine learning on voxels



3D Fully Convolutional Neural Network for Vehicle Detection in Point Clouds

Bo Li, 2017

- Each voxel have only one channel: occupancy value
- Fully 3D convolutional architecture



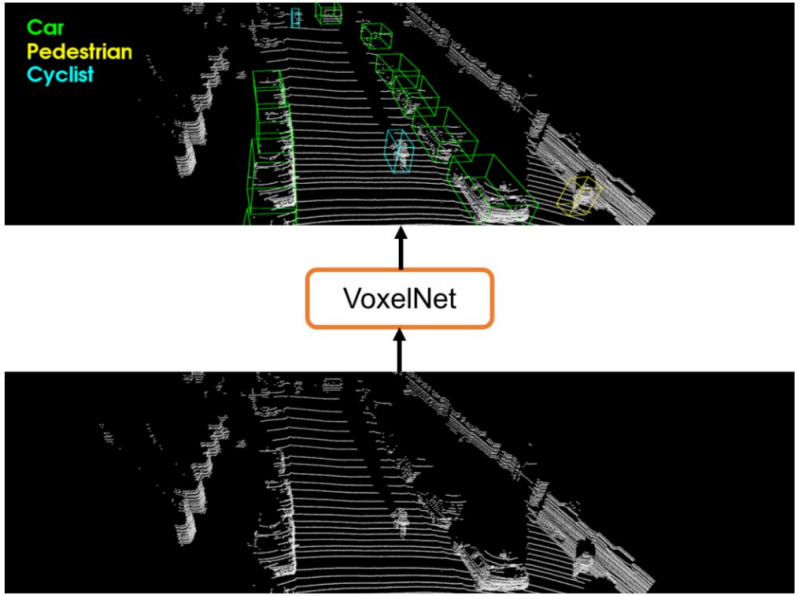
Handcrafted features introduce an information bottleneck, that prevents from effectively exploring 3D shape information

VoxelNet

End-to-End Learning for Point Cloud Based 3D Object Detection

Yin Zhou, Oncel Tuze, Apple Inc 2017

VoxelNet removes the need for manual feature engineering. It learns the descriptive representation by a set of Voxel Feature Encoding layers.

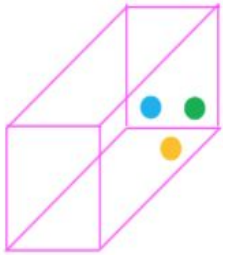


VoxelNet: Voxel Feature Encoding layers (VFE)

End-to-End Learning for Point Cloud Based 3D Object Detection

Yin Zhou, Oncel Tuze, Apple Inc 2017

- A single voxel with 3 points only



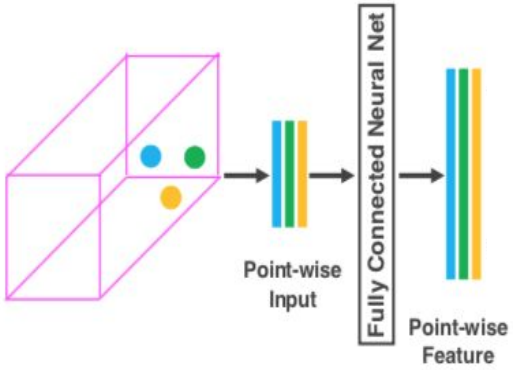
VoxelNet: Voxel Feature Encoding layers (VFE)

End-to-End Learning for Point Cloud Based 3D Object Detection

Yin Zhou, Oncel Tuze, Apple Inc 2017

- Transform point wise input through the fully connected network into a higher dimensional feature space

- The FCN is just a linear, batch normalization and ReLu layer



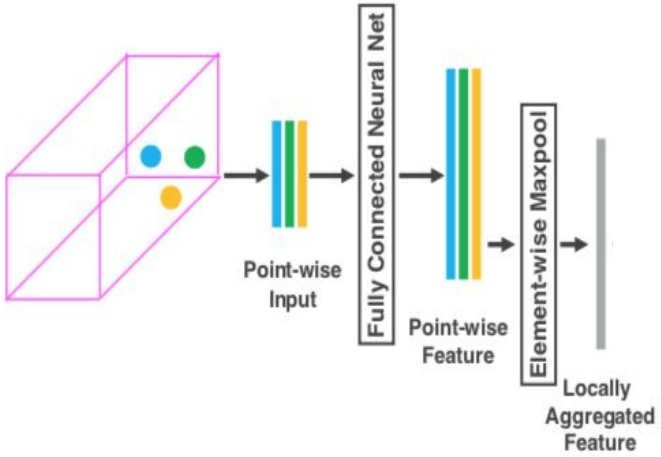
VoxelNet: Voxel Feature Encoding layers (VFE)

End-to-End Learning for Point Cloud Based 3D Object Detection

Yin Zhou, Oncel Tuze, Apple Inc 2017

- Apply element wise max-pooling across all the features to get locally aggregated features

- This captures interactions between points

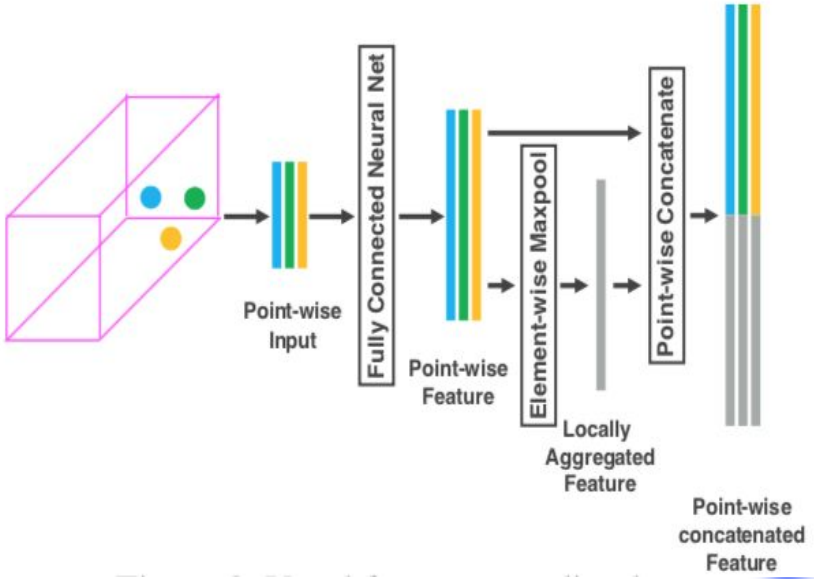


VoxelNet: Voxel Feature Encoding layers (VFE)

End-to-End Learning for Point Cloud Based 3D Object Detection

Yin Zhou, Oncel Tuze, Apple Inc 2017

- Final representations is obtained by combining locally aggregated features and pointwise features

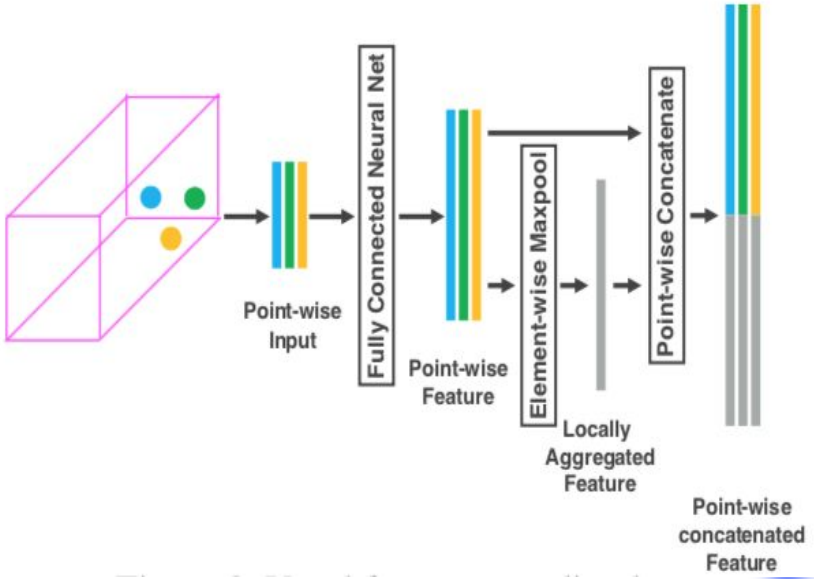


VoxelNet: Voxel Feature Encoding layers (VFE)

End-to-End Learning for Point Cloud Based 3D Object Detection

Yin Zhou, Oncel Tuze, Apple Inc 2017

- Stacking multiple VFE allows for exploring the 3D space effectively and finds a descriptive volumetric representation



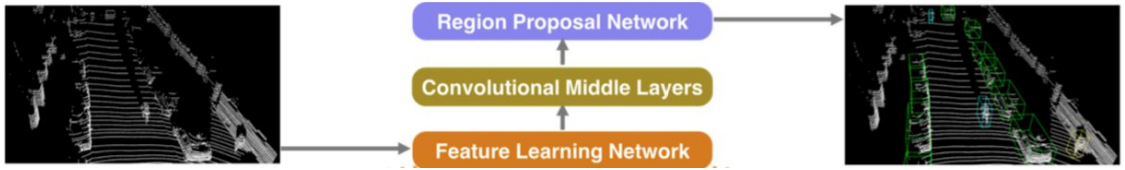
VoxelNet: Voxel Feature Encoding layers (VFE)

End-to-End Learning for Point Cloud Based 3D Object Detection

Yin Zhou, Oncel Tuze, Apple Inc 2017

- 3D convolutions further aggregates local voxel features into high dimensional volumetric representation

- RPN makes final predictions

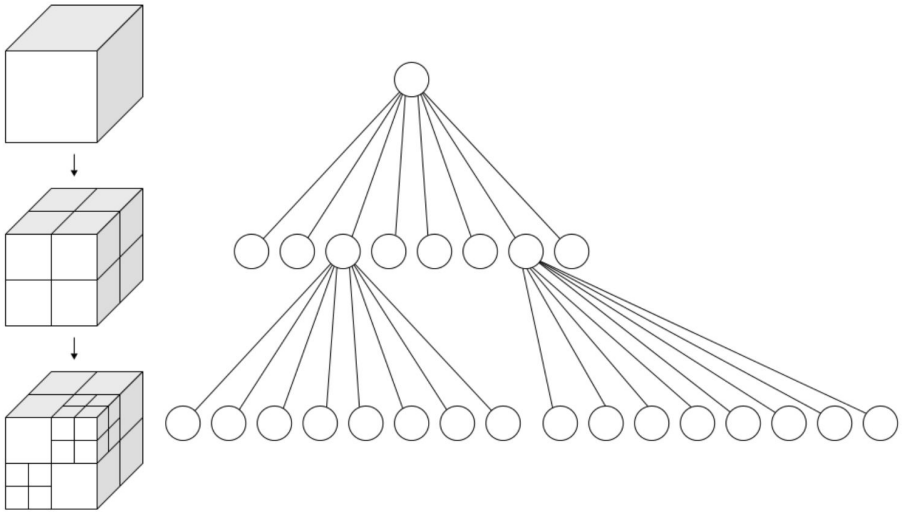


More than 90% of voxels are typically empty

Octree

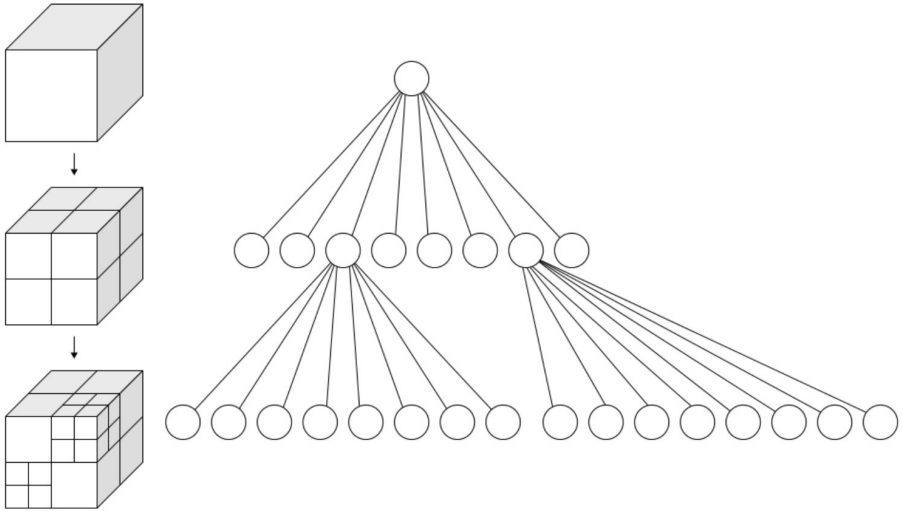
A tree data structure used to recursively subdivide the 3D space into octans.

Simply, just a tree where each internal node has 8 children. Each node represents a subdivision of the space.



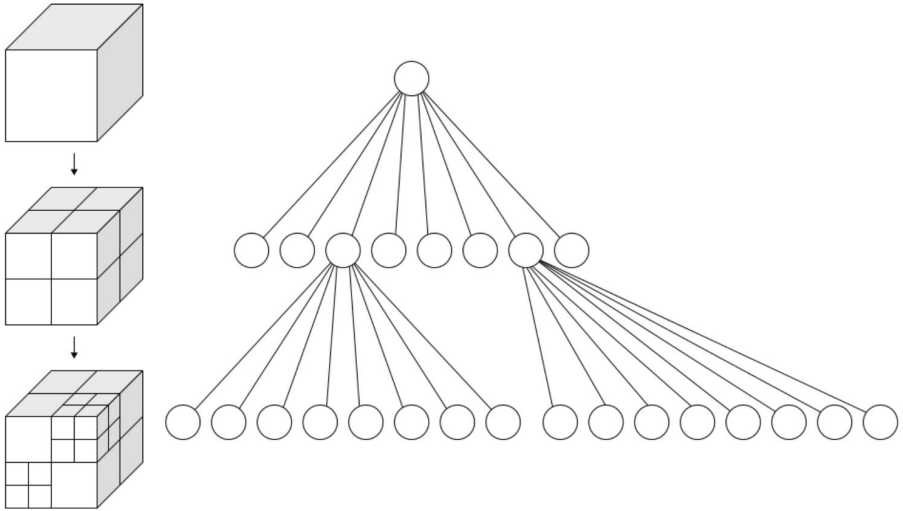
Octree

A single cube contains the whole 3D space - it's represented by a single node.



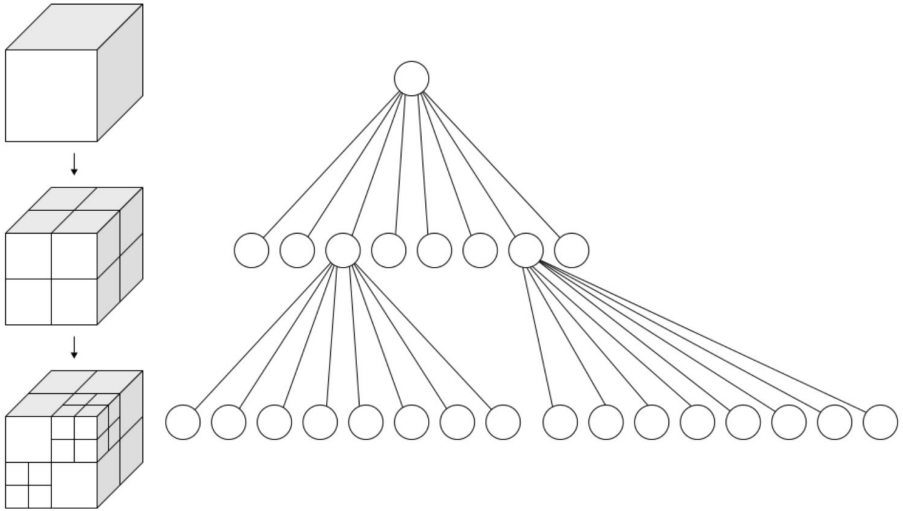
Octree

The initial space was divided into 8-subspaces represented by 8 nodes (octan). The nodes are children of the root.



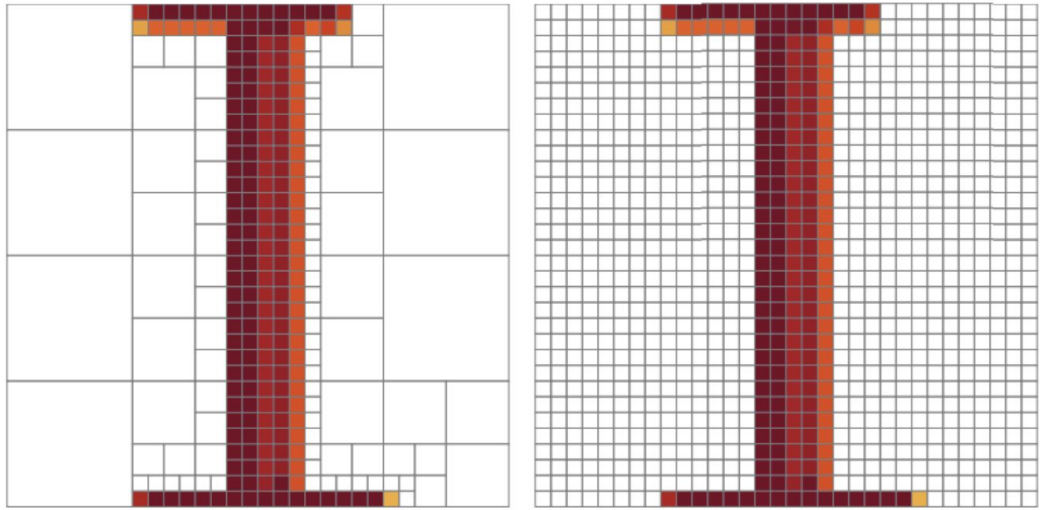
Octree

We keep the process of subdividing the space until there are no more elements left or desired depth is reached



Octree

- Dense representation inefficient: numbers of empty voxels
- Octree keeps information only on non empty voxels
- Very effective representation! No need for memory and computation

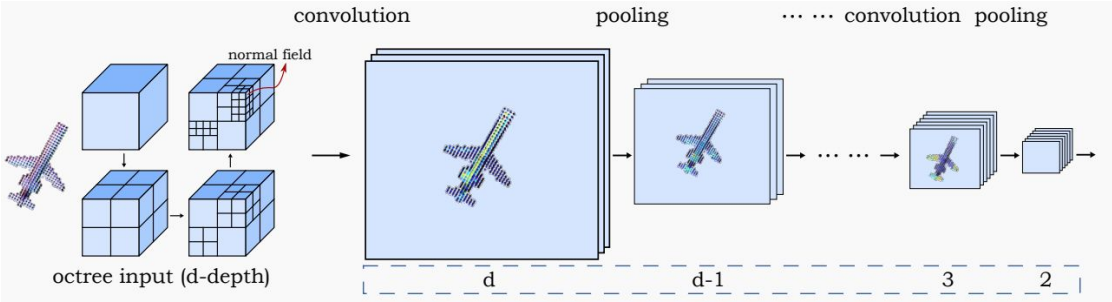


OCNN

Octree-based Convolutional Neural Networks for 3D Shape Analysis

P. Wang, Y. Liu, Y. Guo, C. Sun, X. Tong, SIGGRAPH 2017

- Defines convolution and pooling on octrees
- Octree breaks the spatial neighbourhood of all the voxels
- This neighbourhood needs to be maintained to run the convolutions: hash table

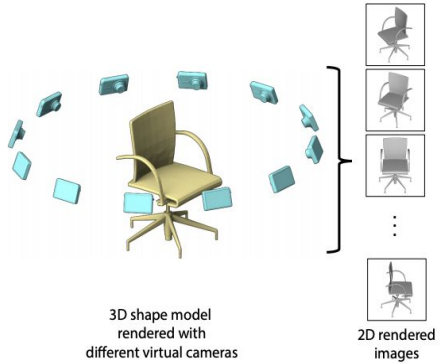


If we managed to move the problem into 2D we would be able to use state-of the art deep learning designed for images

Multiview Convolutional Neural Networks for 3D Shape Recognition

H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, ICVV 2015

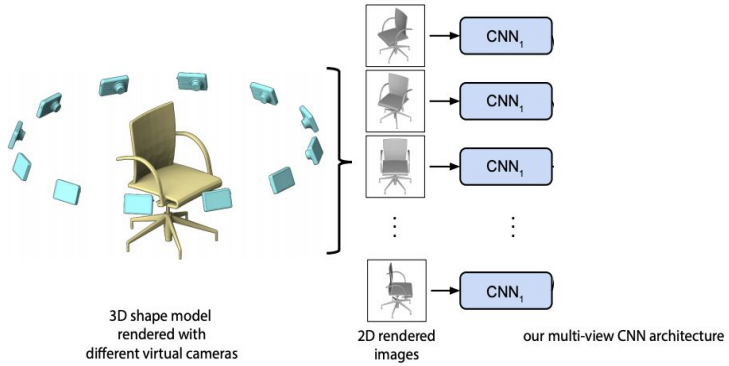
- 12 rendered views by placing
  12 virtual cameras
  around the object every 30°



Multiview Convolutional Neural Networks for 3D Shape Recognition

H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, ICVV 2015

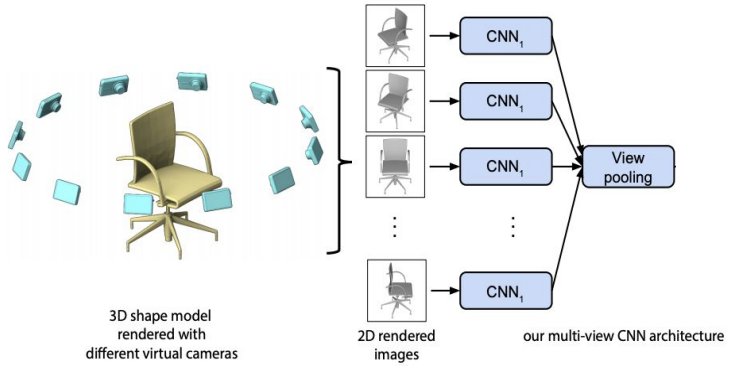
- Each image is passed through the first part of the network (CNN1)
- All branches in the first part of the network share the same parameters in CNN1



Multiview Convolutional Neural Networks for 3D Shape Recognition

H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, ICVV 2015

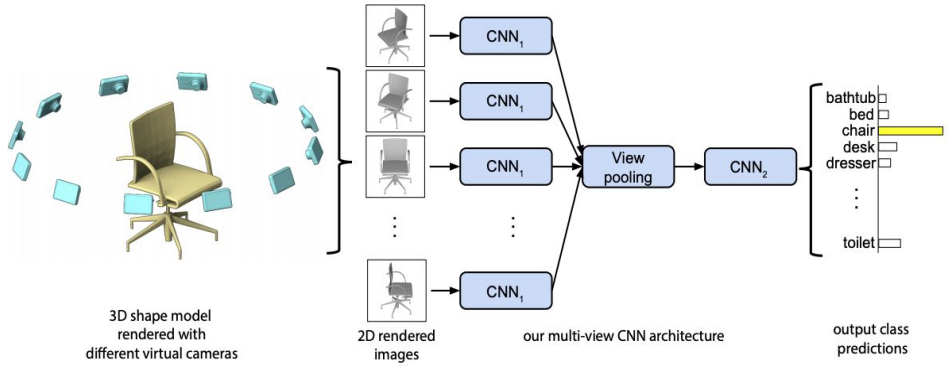
- Aggregate the information from different views by element wise maximum operation across the views



Multiview Convolutional Neural Networks for 3D Shape Recognition

H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, ICCV 2015

- Sent the aggregated features through the rest of the network



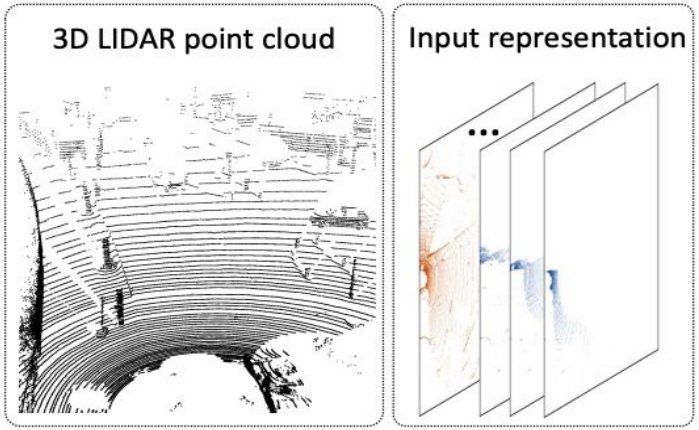
For the autonomous driving scenarios it’s common to use bird eye view projection

Pixor

Real-time 3D Object Detection from Point Clouds

Bin Yang, Wenjie Luo, Raquel Urtasun, Uber Advanced Technologies Group, CVPR 2018

"By reducing the free degree from 3 to 2 we don't lose information as we can still keep the height information as channels along the 3rd dimension (like the RGB channels of 2D images). However, effectively we get a more compact representation since we can apply 2D convolutions to the BEV representation"



Pixor

Real-time 3D Object Detection from Point Clouds

Bin Yang, Wenjie Luo, Raquel Urtasun, Uber Advanced Technologies Group, CVPR 2018

RGB image dimensions: *Width x Height x Channels*

Voxels grid dimensions: *Length x Width x Height x Channels*

Single channel voxels grid dimensions: *Length x Width x Height x* ***1***

Pixor

Real-time 3D Object Detection from Point Clouds

Bin Yang, Wenjie Luo, Raquel Urtasun, Uber Advanced Technologies Group, CVPR 2018

RGB image dimensions: *Width x Height x Channels*

Voxels grid dimensions: *Length x Width x Height x Channels*

Single channel voxels grid dimensions: *Length x Width x Height x 1*

Pixor

Real-time 3D Object Detection from Point Clouds

Bin Yang, Wenjie Luo, Raquel Urtasun, Uber Advanced Technologies Group, CVPR 2018

Single channel voxels grid dimensions:

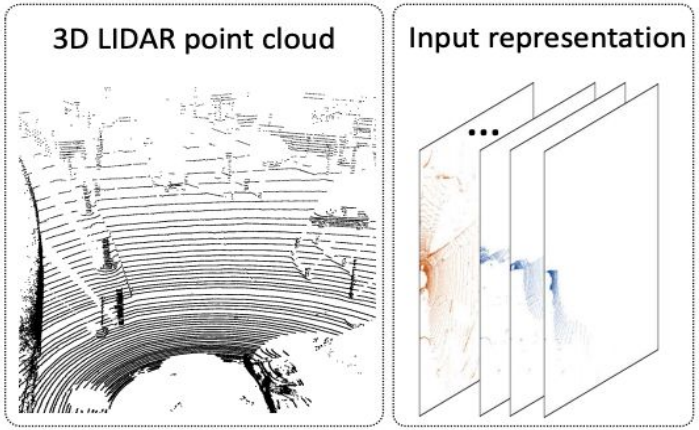
Length x Width x ***Height*** *=* ***Length x Width x Channels*** **= 2D Image!!**

Pixor

Real-time 3D Object Detection from Point Clouds

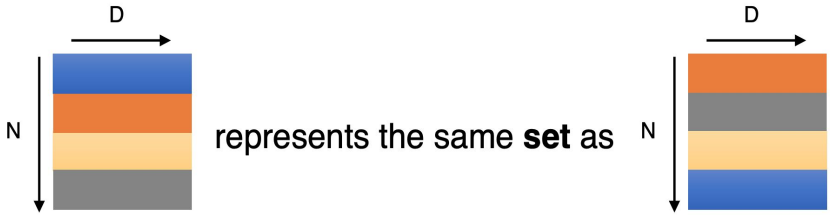
Bin Yang, Wenjie Luo, Raquel Urtasun, Uber Advanced Technologies Group, CVPR 2018

As input is represented now as 2D image with channels, we may use object detection approaches which proved to be effective, such as Single Shot Detectors.



Can we consider raw point clouds as inputs?

Deep learning on raw point clouds



Input permutations invariance

- Mathematically a neural network is just a function
- Permutation invariant neural network corresponds to a function which is symmetric
- In symmetric function any ordering of the input arguments result in always the same value

$$f(x_1, x_2, \ldots, x_n) \overset{def}{=} f(x_{\pi 1}, x_{\pi 2}, \ldots, x_{\pi n})$$

Examples:

$$f(x_1, x_2, \ldots, x_n) = max\{x_1, x_2, \ldots, x_n\}$$
$$f(x_1, x_2, \ldots, x_n) = x_1 + x_2 + \ldots + x_n$$

How can we construct a symmetric function by a neural network?

$$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), h(x_2), \ldots, h(x_n))$$

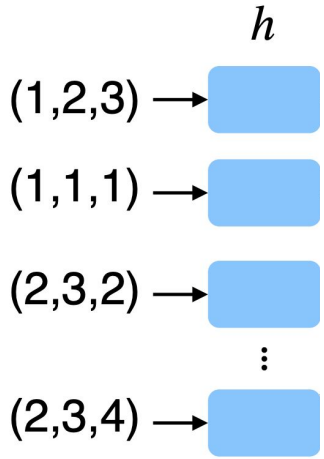
Neural network is symmetric if we manage to express it the following way where g is a symmetric function

“Symmetric” Neural Network 1/3

$$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), h(x_2), \ldots, h(x_n))$$

Transform each point identically and independently by a small network h.

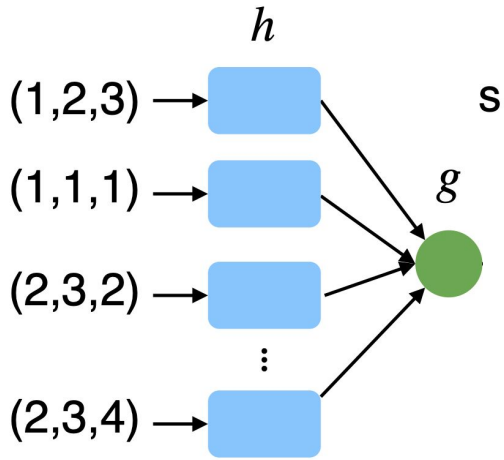
This network will project raw point clouds into high dimensional embedding.



“Symmetric” Neural Network 2/3

$$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), h(x_2), \ldots, h(x_n))$$

Aggregate points embedding by a simple symmetric function

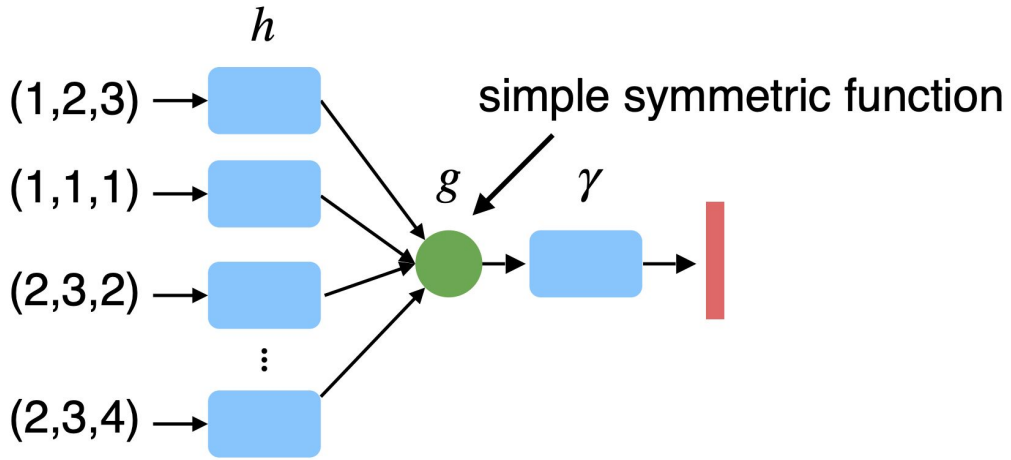


simple symmetric function

“Symmetric” Neural Network 3/3

$$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), h(x_2), \ldots, h(x_n))$$

Post transform the aggregated information by another network γ

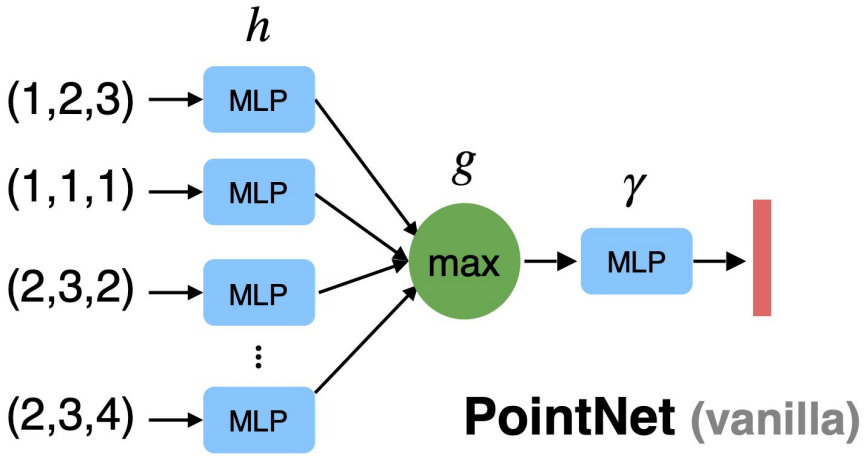


PointNet

Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi, Hao Su, Keichun Mo, Leonidas J. Guibas, Stanford University, CVPR 2017

Empirically they use Multi Layer Perceptron networks for h and γ and Max Pooling as g

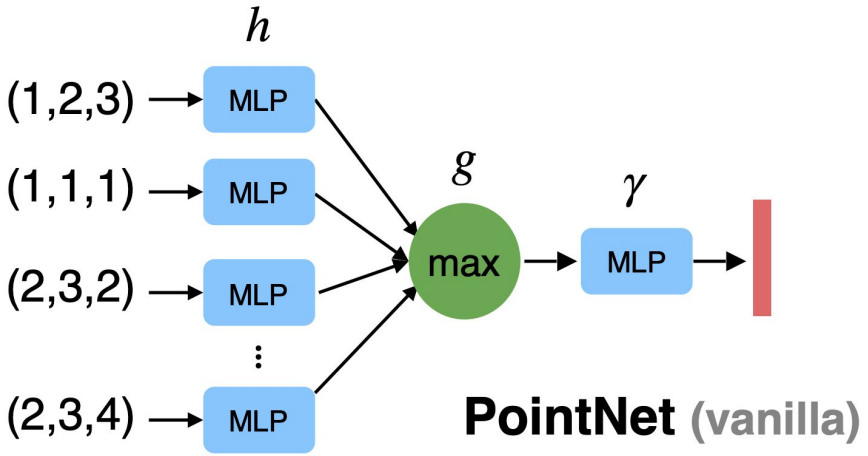


PointNet

Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi, Hao Su, Keichun Mo, Leonidas J. Guibas, Stanford University, CVPR 2017

This network is guaranteed to be invariant to input order



Point clouds rotations should not alter classification results!

How can we construct a model which is invariant under geometric transformations?

Key Idea: align input points clouds into canonical space

Such alignment might be obtained by applying affine transformation to input coordinates

How can we construct a model which is invariant under geometric transformations?

Bad news: we are not able to **define** such alignment for a general case

How can we construct a model which is invariant under geometric transformations?

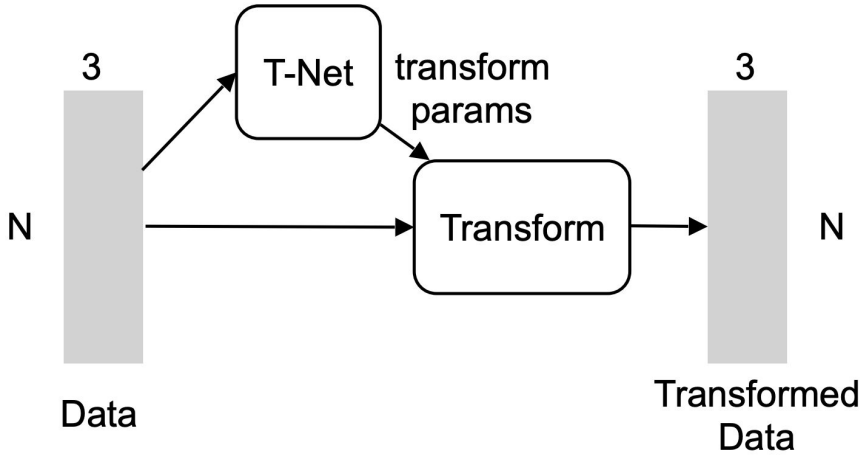
Good news: but we can try to predict it!

PointNet

Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi, Hao Su, Keichun Mo, Leonidas J. Guibas, Stanford University, CVPR 2017

- The prediction is made by a mini pointnet call T-Net
- T-Net is end to end trained with the rest of the network

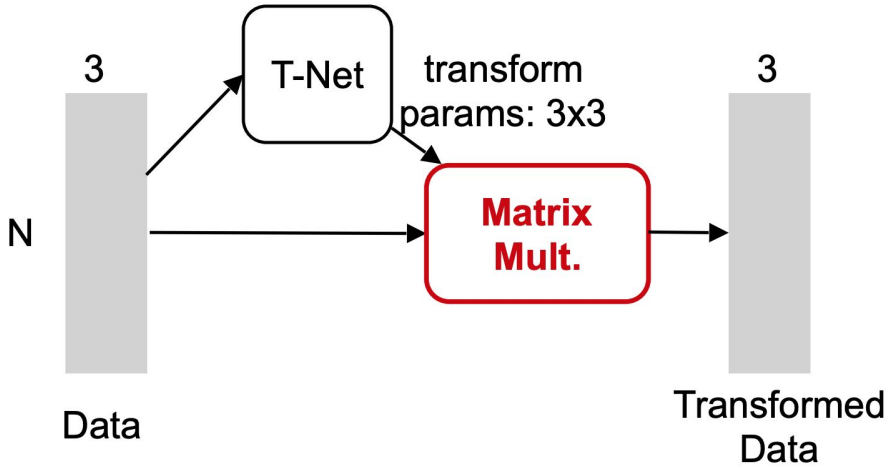


PointNet

Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi, Hao Su, Keichun Mo, Leonidas J. Guibas, Stanford University, CVPR 2017

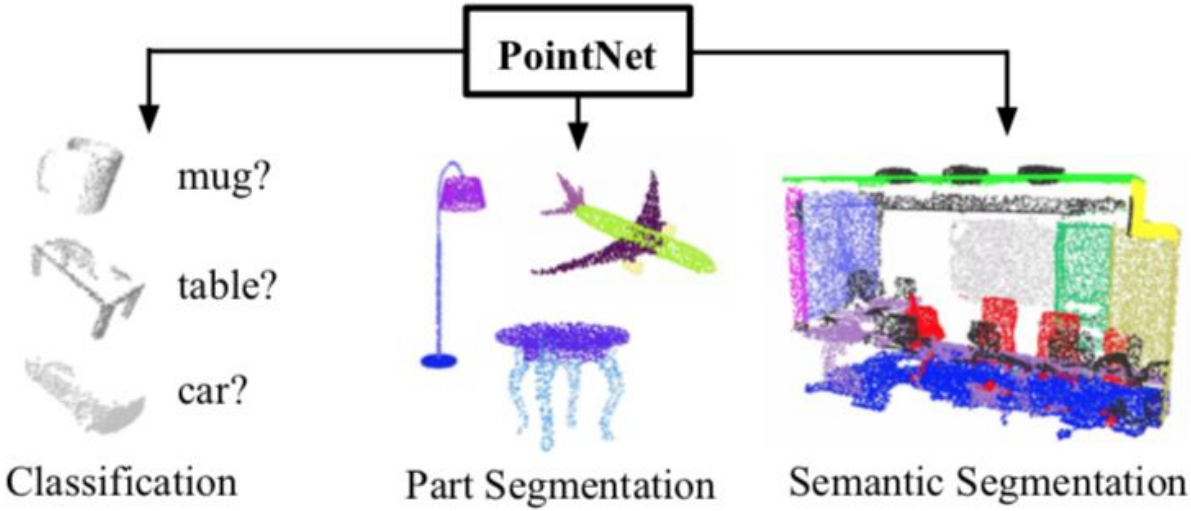
Geometric transformation is just matrix multiplication: **very simple and efficient!**



PointNet

Deep Learning on Point Sets for 3D Classification and Segmentation

Charles R. Qi, Hao Su, Keichun Mo, Leonidas J. Guibas, Stanford University, CVPR 2017



How can we train and evaluate such models?

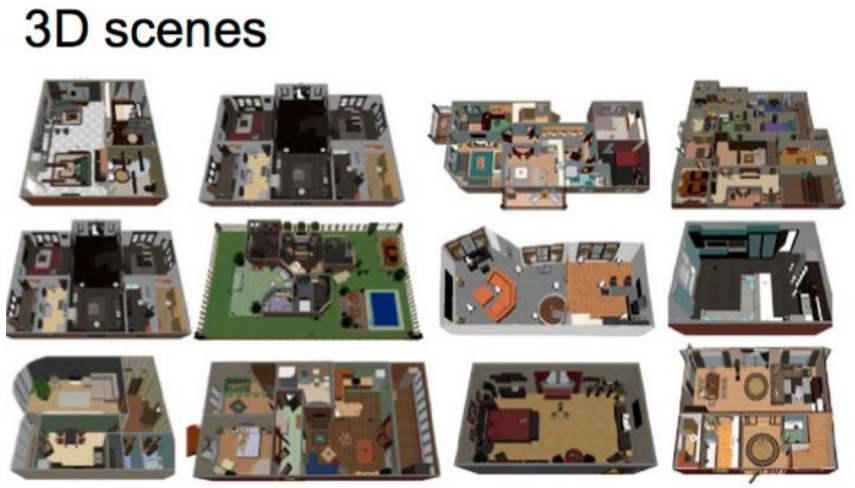
Datasets

- For a long time there was basically a very limited amount of available data for training
- Luckily, right now there are plenty of datasets models, which be used to different tasks of 3D machine learning

- Princeton Shape Benchmark: ~2k 3D polygonal models from world wide web
- ModelNet: 128k models of 662 categories
- ShapeNet: 51k models of 55 categories

Datasets

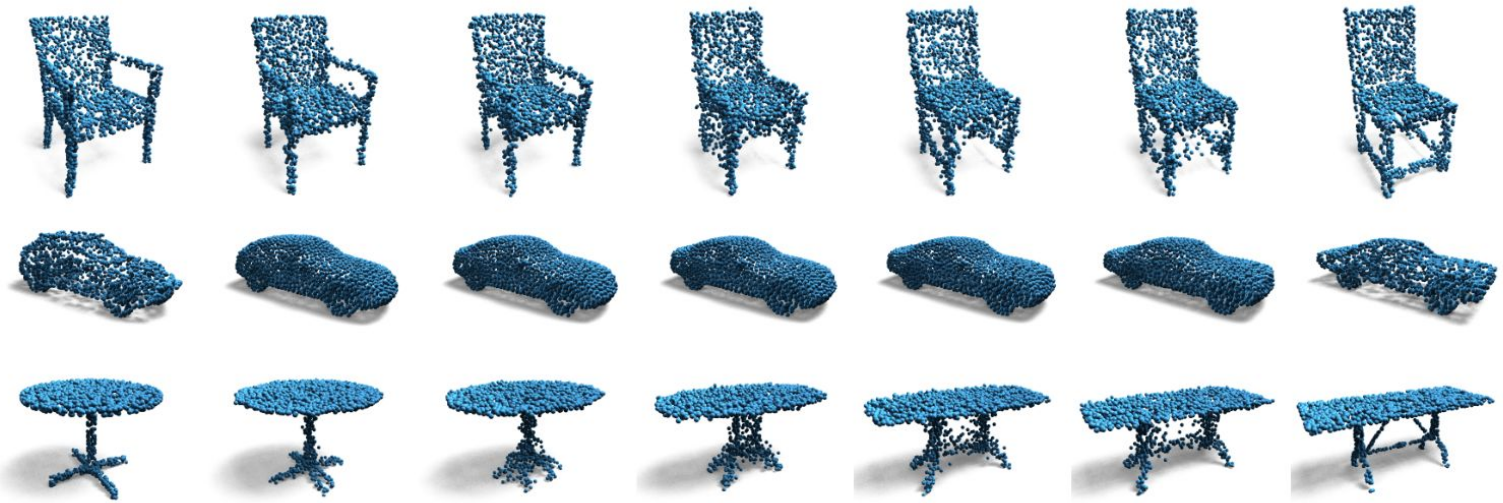
- KITTI: ~80k annotated objects recorded while driving in rural areas of Karlsruhe. It contains ~7.5k lidar scans for training
- Facebook 3D House: 45k indoor 3D scenes
- And many, many more...



Very rapid development of models for SD

No.	Model name	Data of submission
1.	MMF	**November** 2018
2.	ContDen	**November** 2018
3.	Alibaba	October 2018
4.	HDNet	September 2018
5.	Point Pillars	**November** 2018

Very rapid development of models



Adversarial Autoencoders for Generating 3D Point Clouds

M. Zamorski, M. Zięba, R. Nowak, W. Stokowiec, T. Trzciński, Tooploox AI, submission

Summary

+ 3d sensors are becoming more and more common

+ Sensors provide representation in a form of points clouds

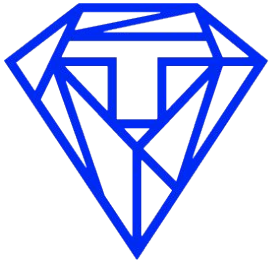
+ Point clouds are unordered sets, where each point might be described by its coordinates

+ Point clouds needs to be tackled properly before using machine learning

Summary

- We may use one of the following techniques to handle point clouds:
  - Voxelization
  - Octree
  - Raw point cloud if proper architecture
  - 2D projection
- There are plenty of publicly available datasets at the moment
- The landscape of 3D deep learning is changing rapidly

We only scratched the surface! There is way more to discover!!



TOOPLOOX

we are hiring!!!

www.tooploox.com

marcin.mosiolek@tooploox.com