Boulder

Computer Vision; 3D

YouTube Playlist

Maziar Raissi

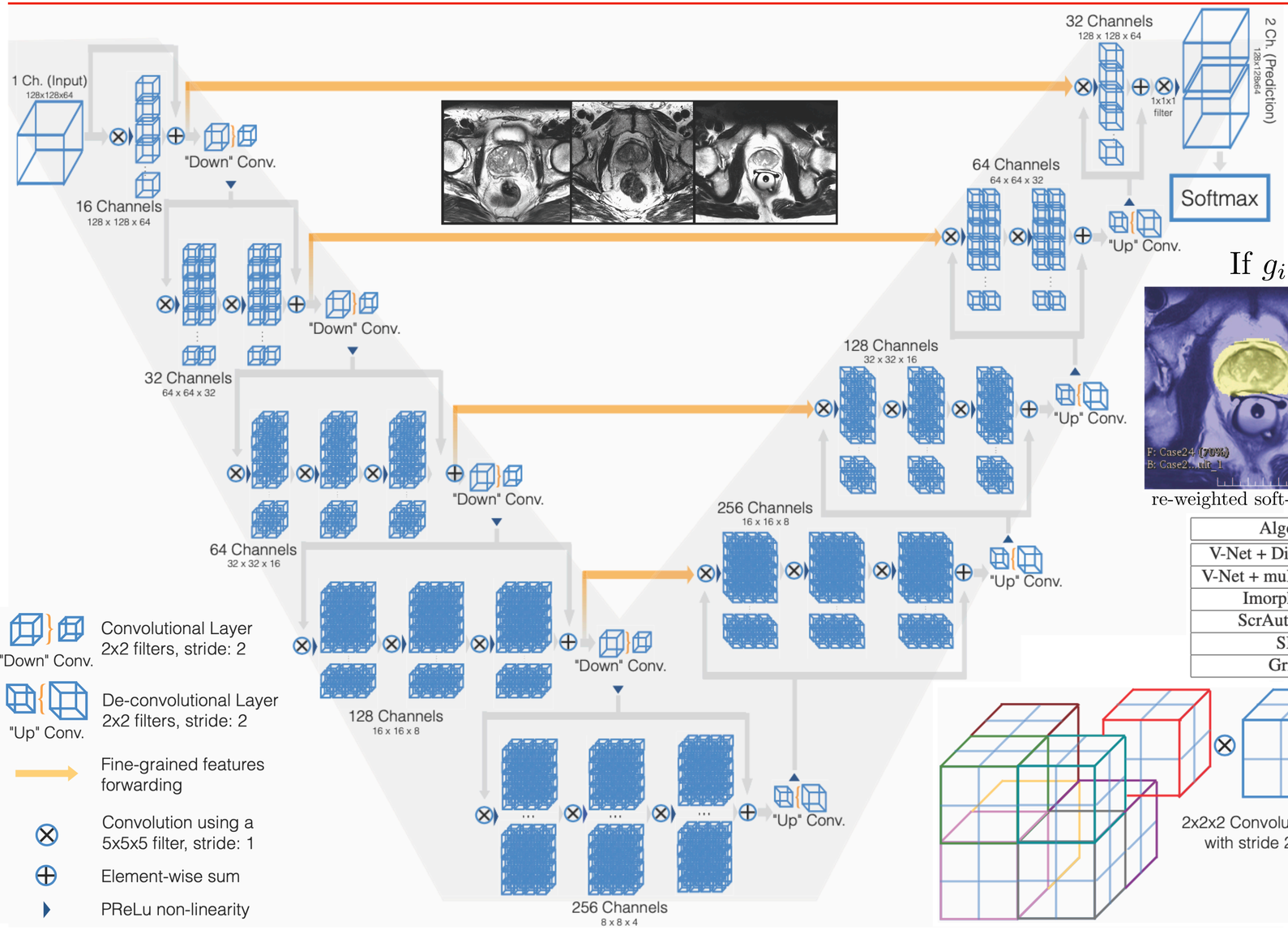
Assistant Professor

Department of Applied Mathematics

University of Colorado Boulder

maziar.raissi@colorado.edu

V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation



Dice Loss

Class Imbalance! (Foreground v.s. Background)

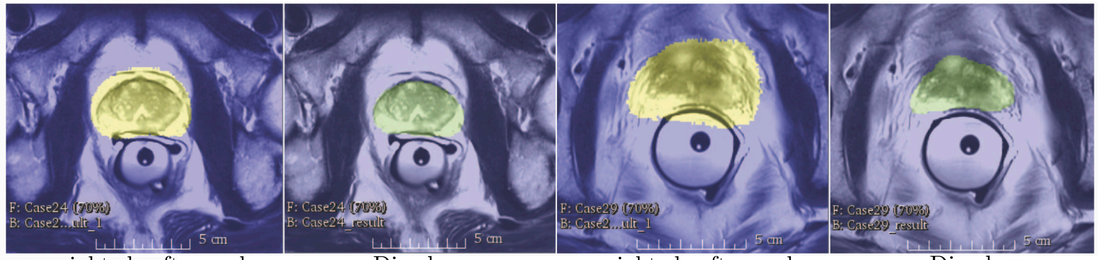
$$D = \frac{2\sum_i^N p_i g_i}{\sum_i^N p_i^2 + \sum_i^N g_i^2}$$

$N \rightarrow$ number of voxels

$p_i \in P \rightarrow$ predicted binary segmentation volume

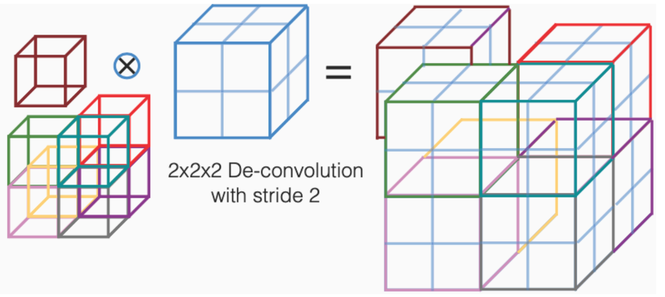
$g_i \in G \rightarrow$ ground truth binary volume

If $g_i = 0$, the background does not contribute to the loss!



re-weighted soft-max loss | Dice loss | re-weighted soft-max loss | Dice loss

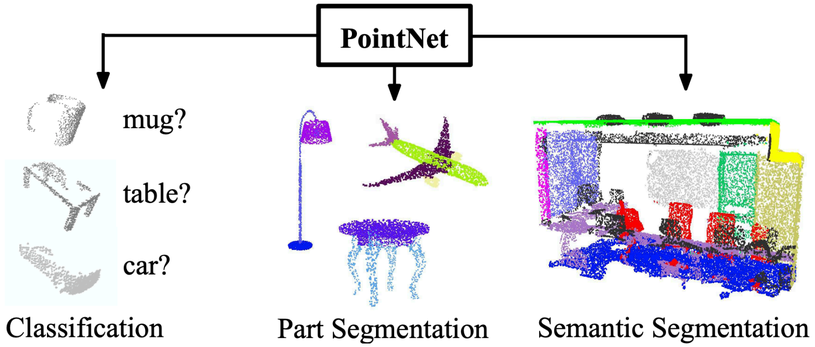
Algorithm	Avg. Dice	Avg. Hausdorff distance	Score on challenge task	Speed
V-Net + Dice-based loss	0.869 ± 0.033	5.71 ± 1.20 mm	82.39	1 sec.
V-Net + mult. logistic loss	0.739 ± 0.088	10.55 ± 5.38 mm	63.30	1 sec.
Imorphics [22]	0.879 ± 0.044	5.935 ± 2.14 mm	84.36	8 min.
ScrAutoProstate	0.874 ± 0.036	5.58 ± 1.49 mm	83.49	1 sec.
SBIA	0.835 ± 0.055	7.73 ± 2.68 mm	78.33	–
Grislies	0.834 ± 0.082	7.90 ± 3.82 mm	77.55	7 min.

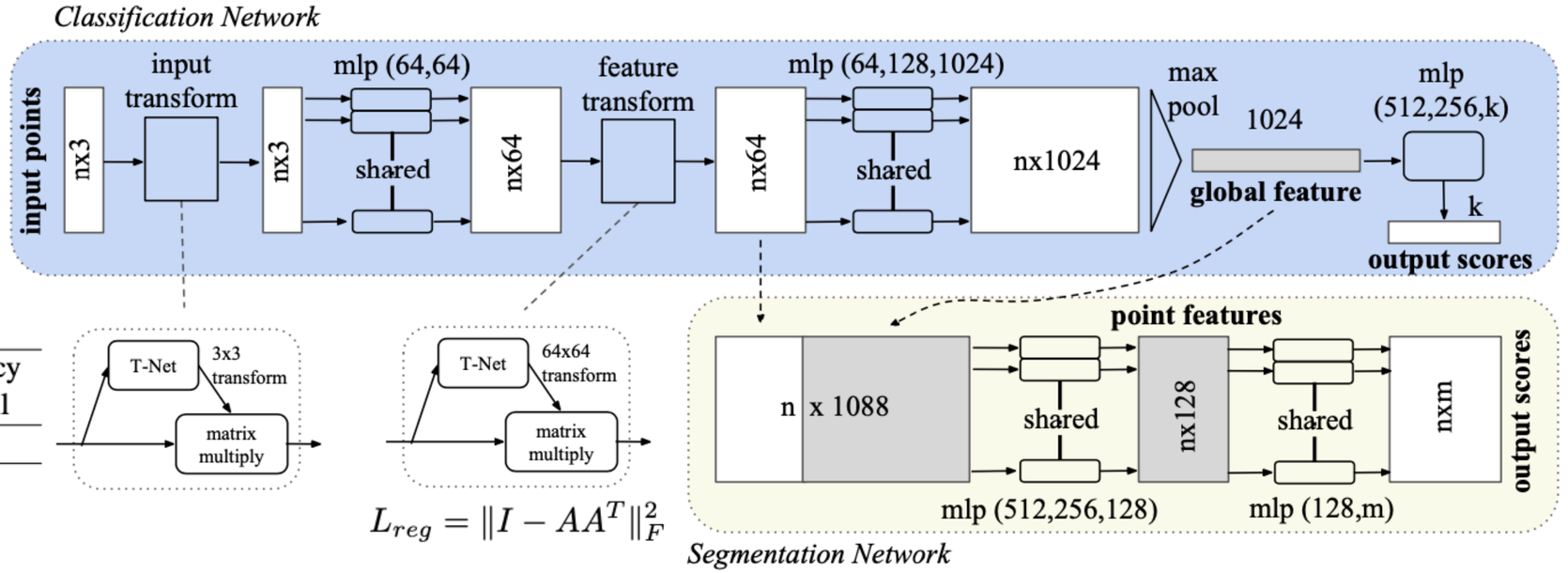


Milletari, Fausto, Nassir Navab, and Seyed-Ahmad Ahmadi. "V-net: Fully convolutional neural networks for volumetric medical image segmentation." *2016 fourth international conference on 3D vision (3DV)*. IEEE, 2016.

PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation

YouTube Video



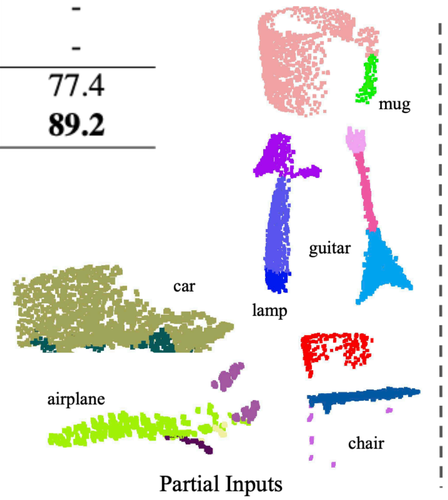


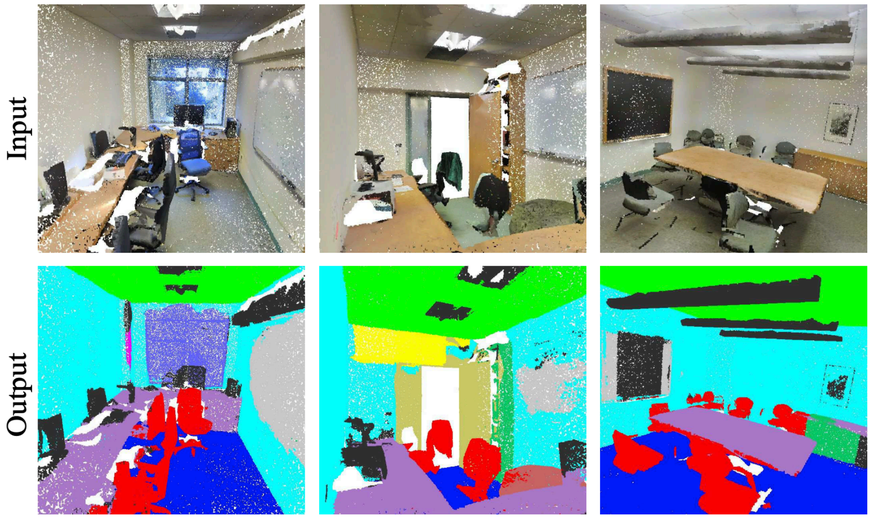
	input	#views	accuracy avg. class	accuracy overall
SPH [11]	mesh	-	68.2	-
3DShapeNets [25]	volume	1	77.3	84.7
VoxNet [15]	volume	12	83.0	85.9
Subvolume [16]	volume	20	86.0	**89.2**
LFD [25]	image	10	75.5	-
MVCNN [20]	image	80	**90.1**	-
Ours baseline	point	-	72.6	77.4
Ours PointNet	point	1	86.2	**89.2**

Classification results on ModelNet40

mIoU(%)	mean	aero	bag	cap	car	chair	ear phone
# shapes		2690	76	55	898	3758	69
Wu [24]	-	63.2	-	-	-	73.5	-
Yi [26]	81.4	81.0	78.4	77.7	**75.7**	87.6	61.9
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5
Ours	**83.7**	**83.4**	**78.7**	**82.5**	74.9	**89.6**	**73.0**

Segmentation results on ShapeNet part dataset





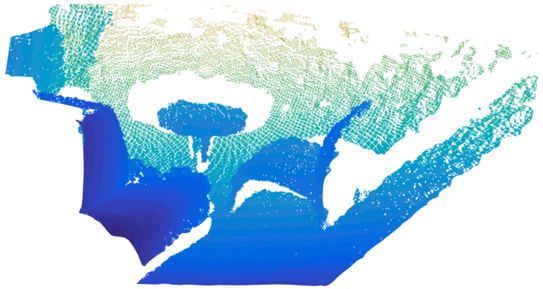
Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space

Boulder

YouTube Video

Visualization of a scan captured from a Structure Sensor



PointNet

$\{x_1, x_2, \ldots, x_n\} \subset \mathbb{R}^d \rightarrow$ unordered point set

$f(x_1, x_2, \ldots, x_n) = \gamma(\max_{i=1,\ldots,n}\{h(x_i)\})$

$\gamma, h \rightarrow$ MLP

Hierarchical Point Set Feature Learning

Local Context

– set abstraction levels
  – sampling layer (centroids of local regions)
  – grouping layer (finding "neighboring" points around centroids)
  – PointNet layer (mini-PointNet)

Sampling Layer

Iterative Farthest Point Sampling (FPS)

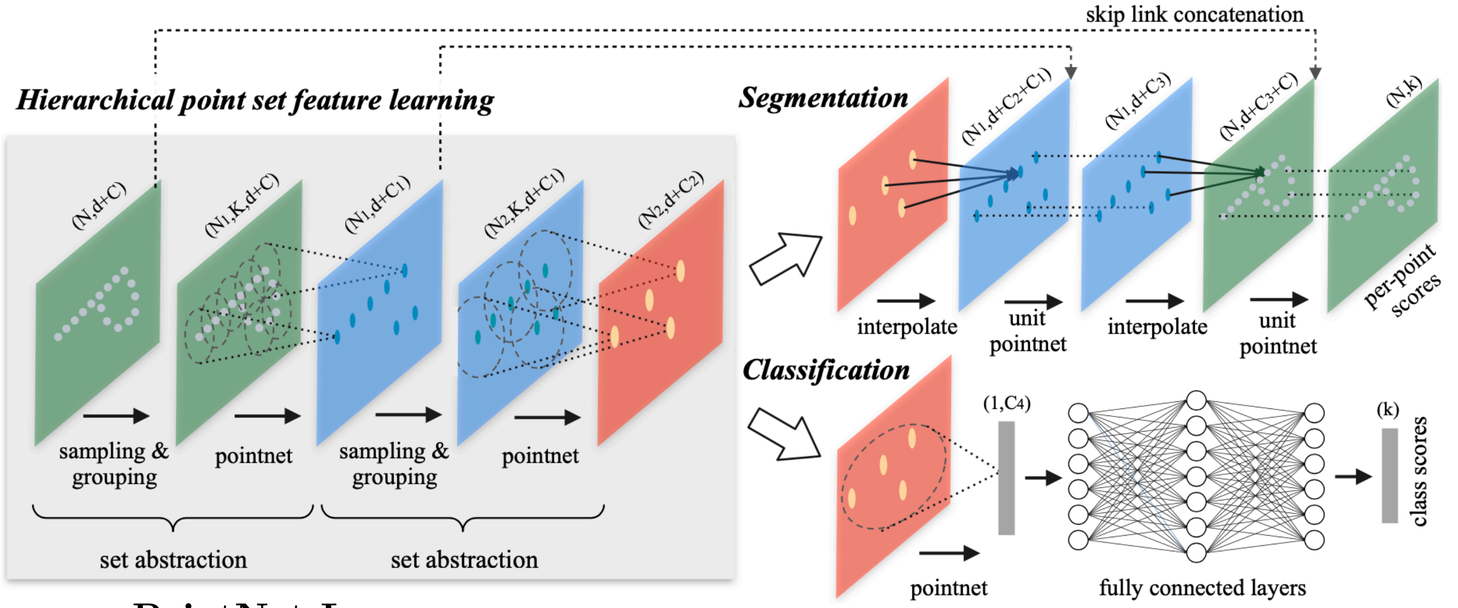
$\{x_{i_1}, x_{i_2}, \ldots, x_{i_m}\} \subset \{x_1, x_2, \ldots, x_n\}$

$x_{i_j} \rightarrow$ the most distant point from the set $\{x_{i_1}, \ldots, x_{i_{j-1}}\}$

Grouping Layer

$K \rightarrow$ number of points in the neighborhood of centroid points

Ball query works better than kNN!



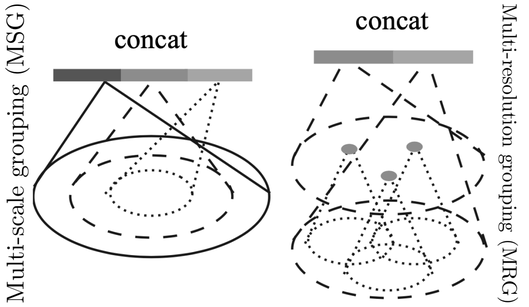
PointNet Layer

Relative coordinates

$x_i^{(j)} \leftarrow x_i^{(j)} - \hat{x}^{(j)}, i = 1, \ldots K, j = 1, \ldots, C$

($\hat{x}$: centroid)

Non-Uniform Sampling Density



Random Input Dropout

For each training point set, choose a dropout ratio $\theta \sim U[0, p]$ (e.g., $p = 0.95$). For each training point, randomly drop the point with probability θ.

Segmentation

$$f^{(j)}(x) = \frac{\sum_{i=1}^{k} w_i(x) f_i^{(j)}}{\sum_{i=1}^{k} w_i(x)}$$

$$\text{where} \quad w_i(x) = \frac{1}{d(x, x_i)^p}, \; j = 1, \ldots, C$$

Inverse Distance Weighted Interpolation averaged based on k nearest neighbors

Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." *Advances in neural information processing systems*. 2017.

Dynamic Graph CNN for Learning on Point Clouds

YouTube Video

EdgeConv → edge convolution
– indoor navigation
– self-driving vehicles
– robotics
– shape synthesis & modeling
LiDAR scanners
– ModelNet40
– ShapeNetPart
– S3DIS

Edge Convolution

$X = \{x_1, \ldots, x_n\} \subset \mathbb{R}^F$
F-dimensional point cloud with n points
$F = 3 \rightarrow x_i = (x_i, y_i, z_i)$
color, surface, normal, etc.
$\mathcal{G} = (\mathcal{V}, \mathcal{E}) \rightarrow$ directed graph representing local point cloud structure
$\mathcal{V} = \{1, \ldots, n\} \rightarrow$ vertices
$\mathcal{E} \subset \mathcal{V} \times \mathcal{V} \rightarrow$ edges
k-nearest neighbor (k-NN) graph of X in $\mathbb{R}^F$
The graph includes self-loop
$e_{ij} = h_\theta(x_i, x_j)$ where $h_\theta : \mathbb{R}^F \times \mathbb{R}^F \rightarrow \mathbb{R}^{F'}$
edge feature
$x'_i = \square_{j:(i,j)\in\mathcal{E}} h_\theta(x_i, x_j) \qquad \square = \sum$ or max

Choice of h & $\square$

$x'_{im} = \sum_{j:(i,j)\in\mathcal{E}} \theta_m \cdot x_j \rightarrow$ standard convolution
m-th filter
$h_\theta(x_i, x_j) = h_\theta(x_i) \rightarrow$ PointNet
$h_\theta(x_i, x_j) = h_\theta(x_j) \rightarrow$ PointCNN
$x'_{im} = \sum_{j\in\mathcal{V}} h_\theta(x_j) g(u(x_i, x_j))$
Gaussian kernel
pairwise distance in Euclidean space
$h_\theta(x_i, x_j) = h_\theta(x_i - x_j) \rightarrow$ encode local information
$h_\theta(x_i, x_j) = \overline{h}_\theta(x_i, x_i - x_j) \rightarrow$ encode global & local information
$e'_{ijm} = \mathrm{ReLU}(\theta_m \cdot (x_j - x_i) + \phi_m \cdot x_i)$
$x'_{im} = \max_{j:(i,j)\in\mathcal{E}} e'_{ijm}$

Dynamic Graph CNN (DGCNN)

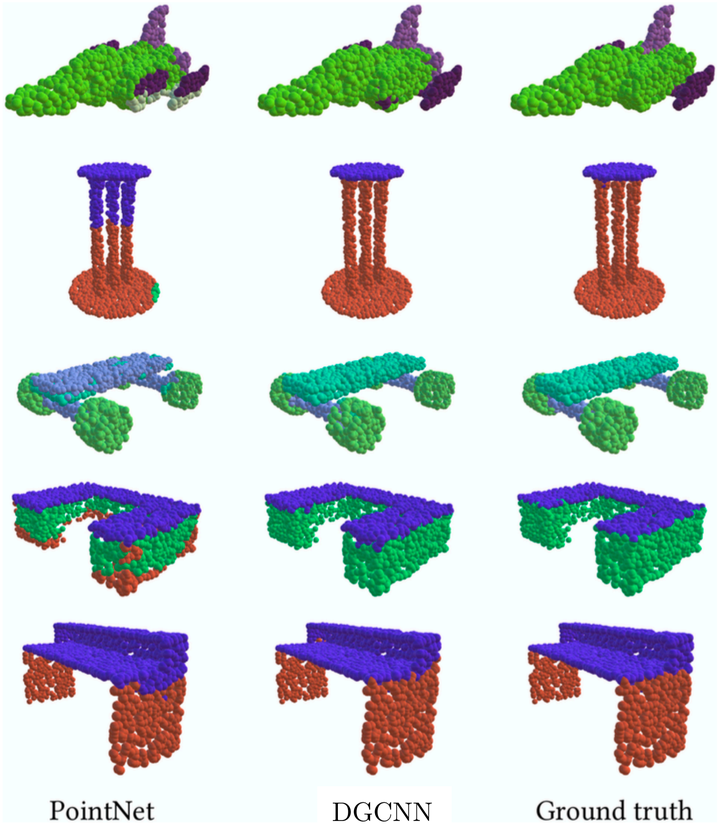
$\mathcal{G}^\ell = (\mathcal{V}^\ell, \mathcal{E}^\ell) \rightarrow$ dynamically constructed per each layer

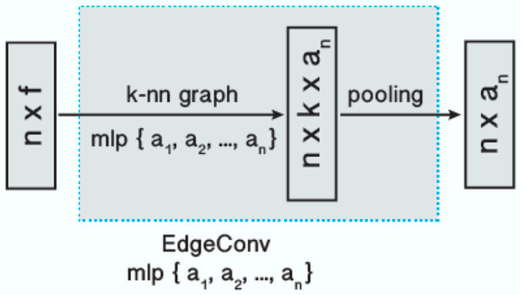
Datasets

ModelNet40 contains 12,311 meshed CAD models from 40 categories.

ShapeNet part dataset contains 16,881 3D shapes from 16 object categories, annotated with 50 parts in total.

Stanford Large-Scale 3D Indoor Spaces Dataset (S3DIS) includes 3D scan point clouds for 6 indoor areas including 272 rooms in total. Each point belongs to one of 13 semantic categories—e.g., board, bookcase, chair, ceiling, and beam–plus clutter





Wang, Yue, et al. "Dynamic graph cnn for learning on point clouds." *Acm Transactions On Graphics (tog)* 38.5 (2019): 1-12.

Point Transformer

– autonomous driving – augmented reality – robotics
S3DIS (Stanford large-scale 3D Indoor Spaces): semantic scene segmentation
ShapeNetPart: object part segmentation
ModelNet40: shape classification

Transformers Background

– scalar attention – vector attention
$\mathcal{X} = \{x_i\} \rightarrow$ set of feature vectors

The set can be a collection of feature vectors that represent the entire signal (e.g., sentence or image) or a collection of feature vectors from a local patch within the signal (e.g., an image patch).

$y_i = \sum_{x_j \in \mathcal{X}} \rho(\varphi(x_i)^T \psi(x_j) + \delta)\alpha(x_j) \rightarrow$ standard scalar dot-product attention layer

$y_i \rightarrow$ output feature
$\varphi, \psi, \alpha \rightarrow$ feature transformations (linear projections or MLPs)
$\delta \rightarrow$ position encoding function
$\rho \rightarrow$ normalization function (e.g., softmax)

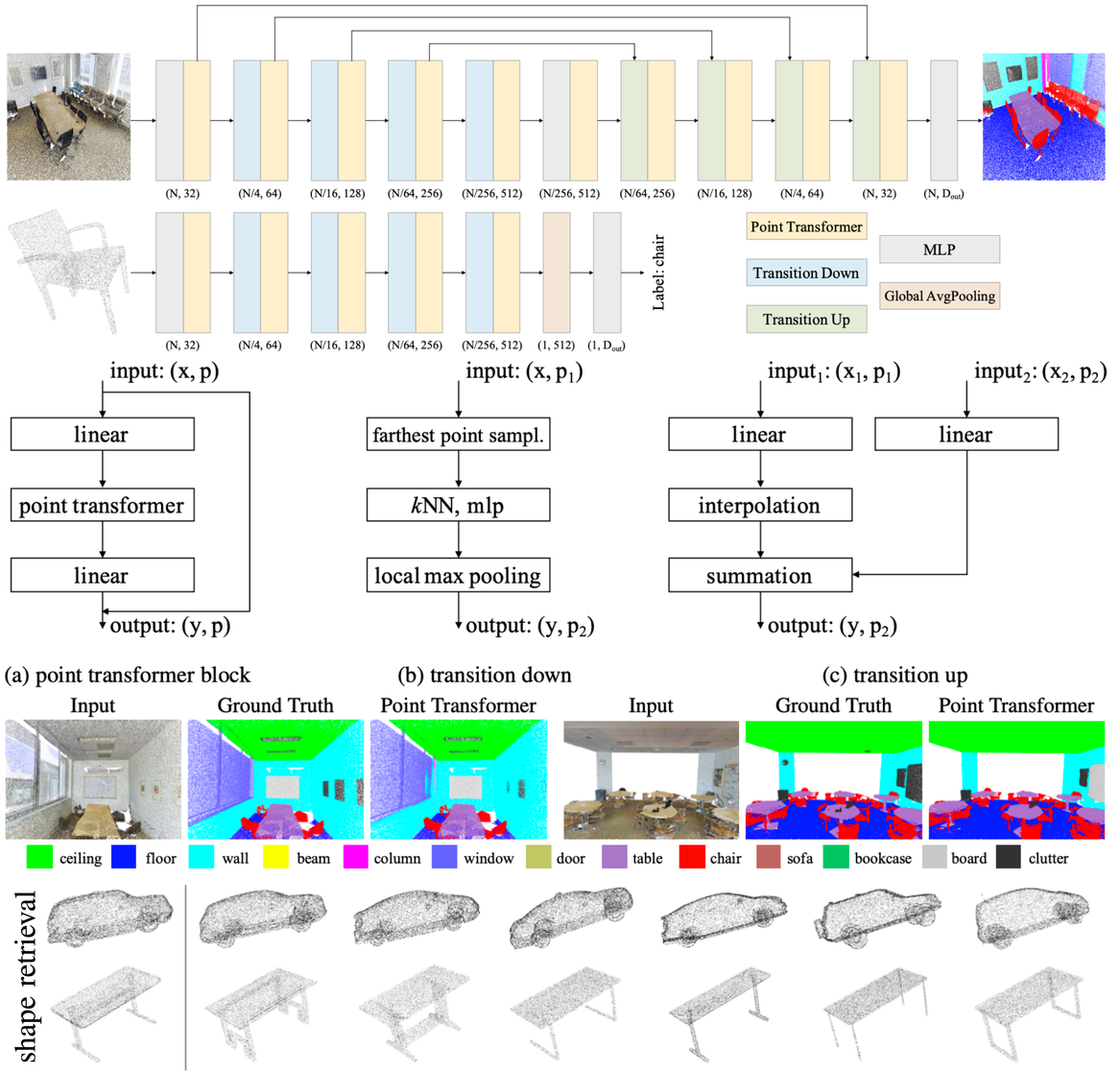
$y_i = \sum_{x_j \in \mathcal{X}} \rho(\gamma(\beta(\varphi(x_i), \psi(x_j)) + \delta)) \odot \alpha(x_j) \rightarrow$ vector attention
(can modulate individual feature channels)

$\beta \rightarrow$ relation function (e.g., subtraction)
$\gamma \rightarrow$ mapping function (e.g., an MLP)

Point Transformer Layer

$$y_i = \sum_{x_j \in \mathcal{X}(i)} \rho(\gamma(\varphi(x_i) - \psi(x_j) + \delta)) \odot (\alpha(x_j) + \delta)$$

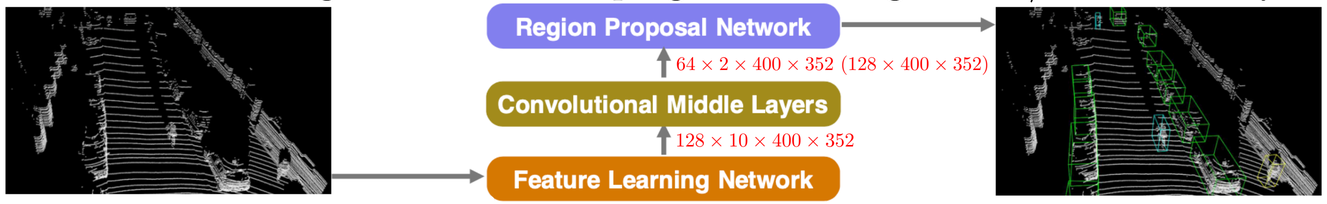
$\mathcal{X}(i) \subset \mathcal{X} \rightarrow$ set of points in a local neighborhood (specifically, k nearest neighbors) of x_i
$\delta = \theta(p_i - p_j) \rightarrow$ position encoding $\quad p_i, p_j \rightarrow$ 3D point coordinates for points i and j $\quad \theta \rightarrow$ MLP (two linear layers and one ReLU nonlinearity)

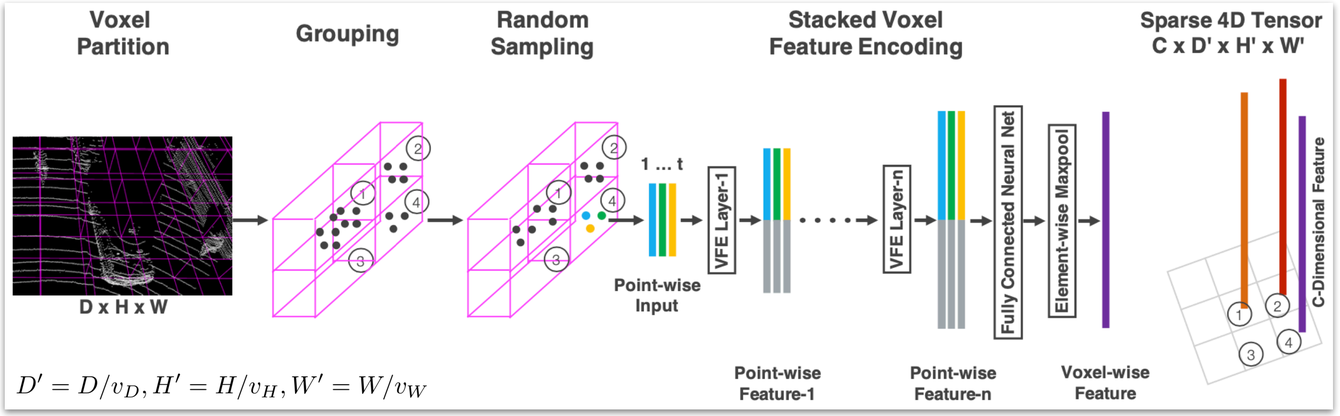


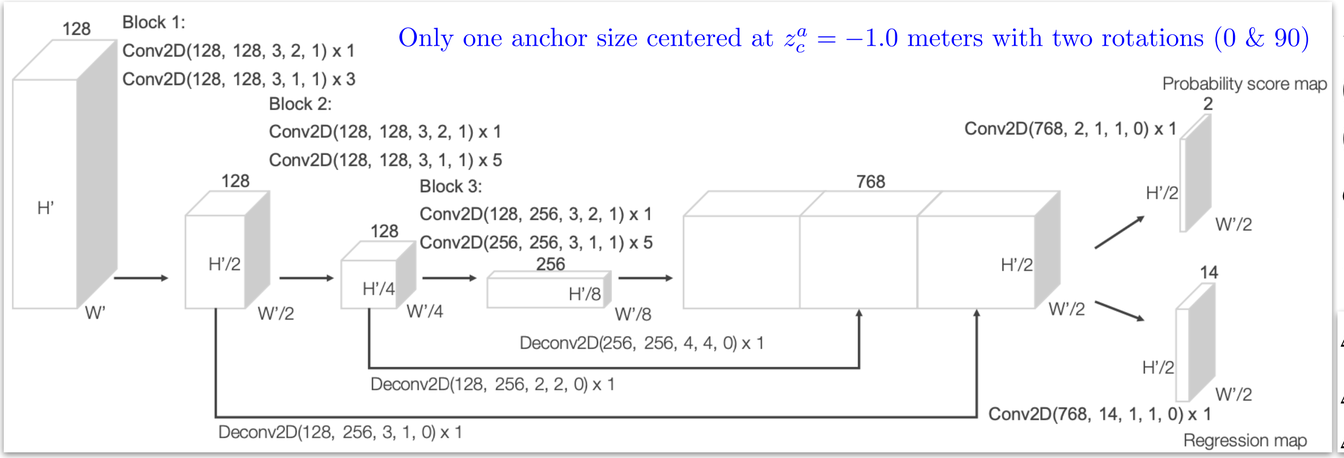
Zhao, Hengshuang, et al. "Point transformer." *arXiv preprint arXiv:2012.09164* (2020).

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

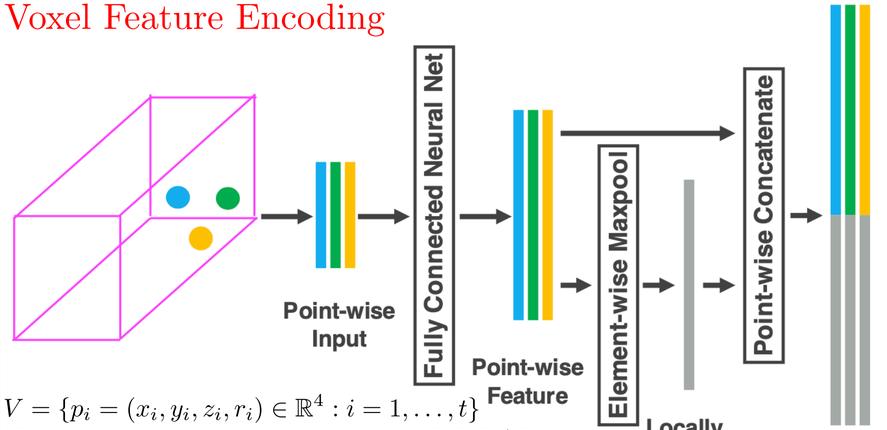
– autonomous navigation – housekeeping robots – augmented/virtual reality

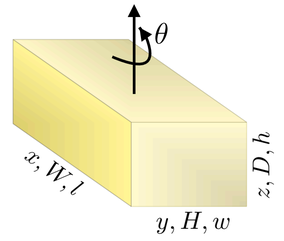






Voxel Feature Encoding





$V = \{p_i = (x_i, y_i, z_i, r_i) \in \mathbb{R}^4 : i = 1, \ldots, t\}$

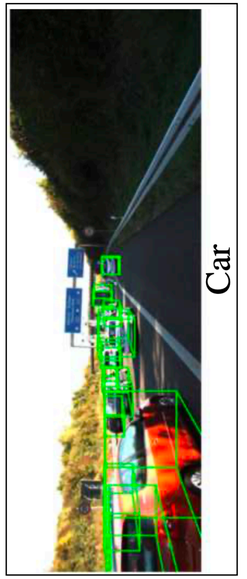
$V \rightarrow$ nonempty voxel containing $t \leq T$ LiDAR points

$r_i \rightarrow$ received reflectance

$(v_x, v_y, v_z) \rightarrow$ local mean as the centroid of all points in V

$V_{\text{in}} = \{\hat{p}_i = (x_i, y_i, z_i, r_i, x_i - v_x, y_i - v_y, z_i - v_z) \in \mathbb{R}^7 : i = 1, \ldots, t\}$

$V_{\text{in}} \rightarrow$ input feature set



Loss Function

$\{a_i^{\text{pos}} : i = 1, \ldots, N_{\text{pos}}\} \rightarrow$ set of positive anchors

$\{a_j^{\text{neg}} : j = 1, \ldots, N_{\text{neg}}\} \rightarrow$ set of negative anchors

$(x_c^g, y_c^g, z_c^g, l^q, w^g, h^g, \theta^g) \rightarrow$ 3D ground truth box

$(x_c^a, y_c^a, z_c^a, l^a, w^a, h^a, \theta^a) \rightarrow$ matching positive anchor

$d^a = \sqrt{(l^a)^2 + (w^a)^2} \rightarrow$ diagonal of the base of the anchor box

$$L = \alpha \frac{1}{N_{\text{pos}}} \sum_i L_{\text{cls}}(p_i^{\text{pos}}, 1) + \beta \frac{1}{N_{\text{neg}}} \sum_j L_{\text{cls}}(p_j^{\text{neg}}, 0) + \frac{1}{N_{\text{pos}}} \sum_i L_{\text{reg}}(\mathbf{u}_i, \mathbf{u}_i^*)$$

$$\Delta x = \frac{x_c^g - x_c^a}{d^a}, \Delta y = \frac{y_c^g - y_c^a}{d^a}, \Delta z = \frac{z_c^g - z_c^a}{h^a},$$

$$\Delta l = \log(\frac{l^g}{l^a}), \Delta w = \log(\frac{w^g}{w^a}), \Delta h = \log(\frac{h^g}{h^a}),$$

$$\Delta\theta = \theta^g - \theta^a$$

$u^* \in \mathbb{R}^7 \rightarrow$ residual vector

Data Augmentation & Efficient Implementation of the Feature Learning Network $\rightarrow$ (see the paper!)

Zhou, Yin, and Oncel Tuzel. "Voxelnet: End-to-end learning for point cloud based 3d object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.

Boulder

Questions?

YouTube Playlist