Deep Learning in 2D

Dr. Connie Ko

42nd Canadian Symposium on Remote Sensing
Understanding Our World: Remote Sensing For A Sustainable Future July 21-24 2021

Housekeeping items

- Workshop will be recorded
  - For registrants who may have lost internet connection
- Sharing pdf notes
- Github: https://github.com/maryumjam/Deep-Learning-Workshop-CSRS-2021

Workshop Agenda

Morning

Lecture: Deep Learning in 2D

Presenter: Connie Ko

Demo: Mask RCNN for Individual Tree Crown Delineation

Presenter: Andrew Chadwick

Afternoon

Lecture: Deep Learning in 3D

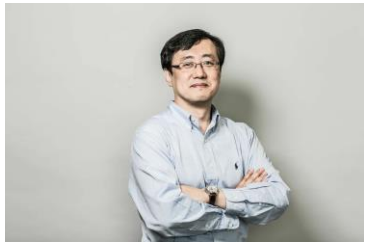
Presenter: Connie Ko

Demo: Deep Learning for 3D Point Cloud - PointNet and 3D-Unet

Presenter: Maryam Jameela

Introduction

https://gunhosohn.me/



Dr. Gunho Sohn

https://gunhosohn.me/team/

Members that are presenting in CSRS 2021

Dr. Connie Ko

Maryam Jameela

Muhammad Kamran

Sunghwan Yoo

Session 2-12a AI in Remote Sensing

Poster Session 2

3DMMAI (https://gunhosohn.me/3dmmai/ ; https://gunhosohn.me/2020/09/06/3d-mobile-mapping-ai-project/)

Our lab will lead the development of deep learning-based

- Object Detection
- Noise Filtering
- Terrain Filtering
- Semantic Segmentation
- Building Footprint Extraction
- LIDAR and visual SLAM development
- Transmission Network Modeling

NSERC CRSNG

TELEDYNE OPTECH Everywhereyoulook™

LASSONDE SCHOOL OF ENGINEERING

YORK UNIVERSITÉ UNIVERSITY

AUSMLab Augmented Urban Space Modeling Lab

GeoICT

Introduction

IRSS: Integrated Remote Sensing Studio
Dr. Nicholas Coops

Andrew Chadwick
https://irsslab.forestry.ubc.ca/people/graduate-students/

Andrew is developing an operational tool that will leverage UAV-derived orthographic imagery and photogrammetric point clouds to deliver enhanced post-harvest regeneration inventory data. This project bridges the burgeoning fields of remote sensing, computer vision, and machine learning.

THE UNIVERSITY OF BRITISH COLUMBIA

Vancouver Campus

Different Architecture of Deep Learning

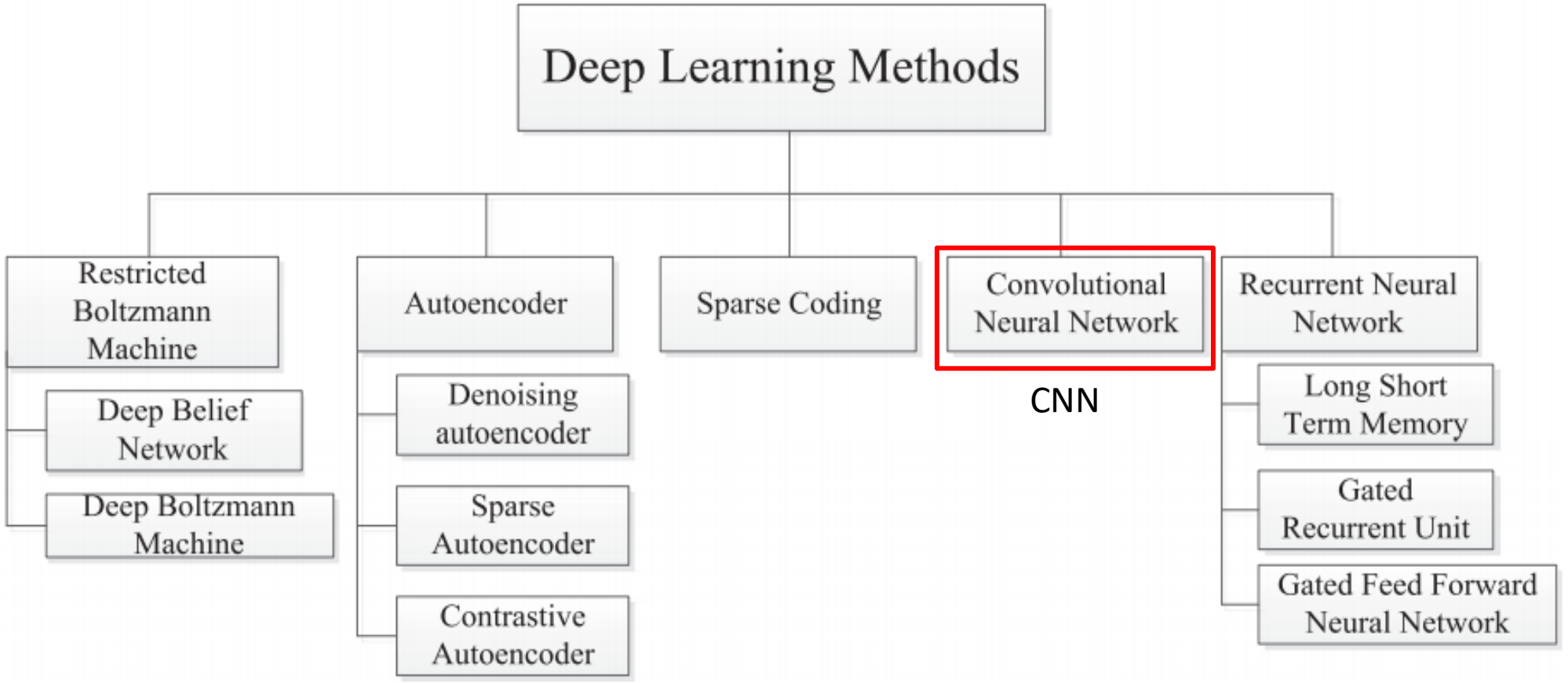


Fig. 1. Different architecture of deep learning algorithms.

Henry Friday Nweke, Ying Wah Teh, Mohammed Ali Al-garadi, Uzoma Rita Alo, Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges, Expert Systems with Applications, Volume 105, 2018, Pages 233-261,

Definitions

- Loss Functions and Regularization
- Optimization
- Convolution Layer
- Pooling Layer
- Activation
- Batch Normalization
- Fully Connected Layers

Common vision tasks:

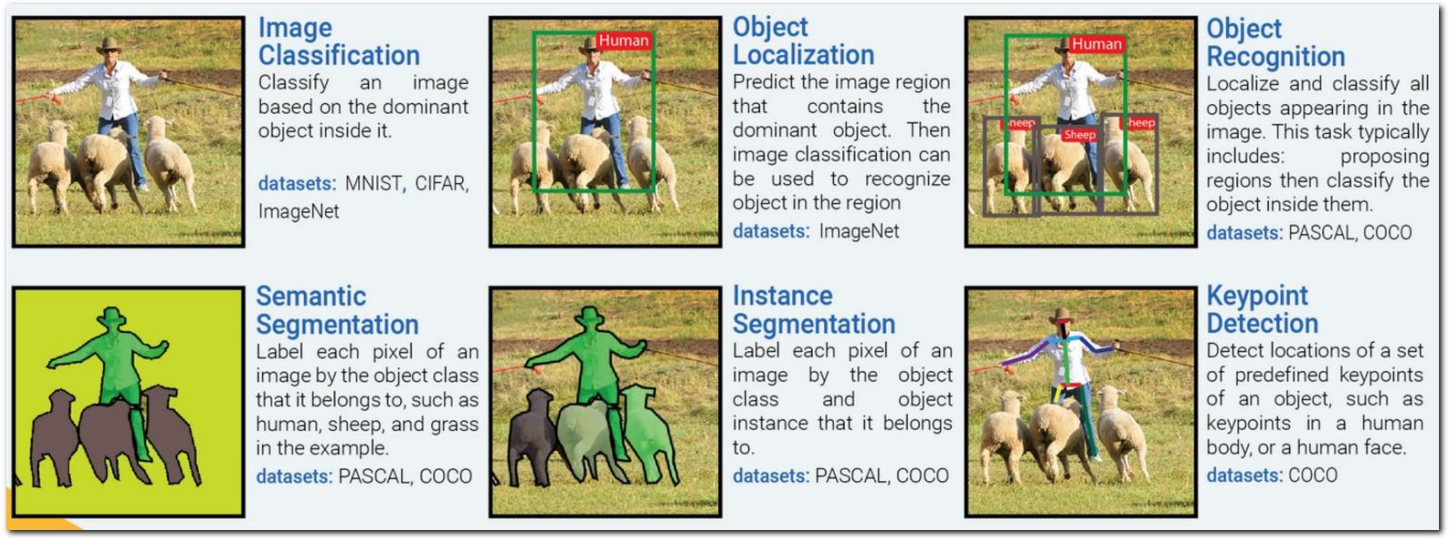
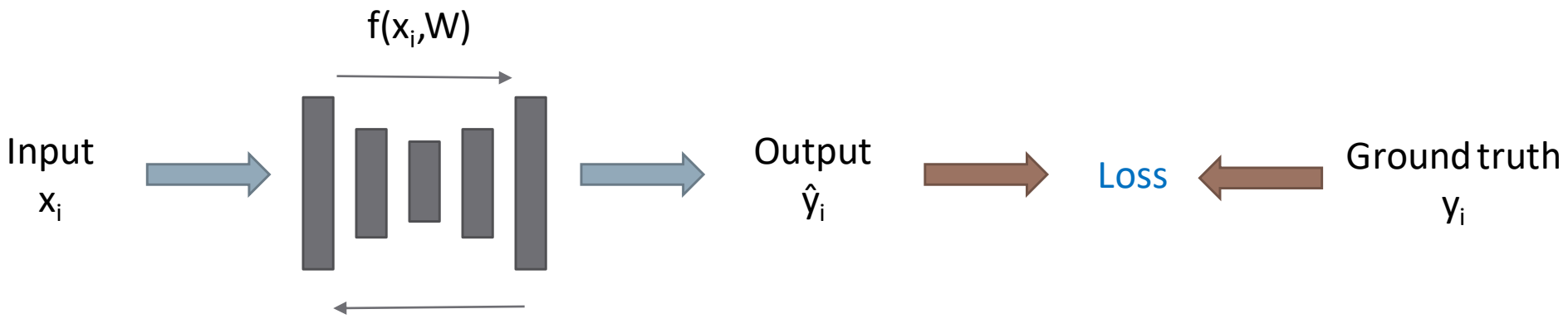


Image source: https://towardsdatascience.com/deep-learning-computer-vision-and-automated-optical-inspection-774e8ca529d3

Loss Functions and Regularization



Loss measures how "bad" the predictions are, therefore, we want to minimize Loss

Dataset (x_i, y_i)
- Training Data
- Validation Data
- Test Data

Loss Functions and Regularization

Multiclass SVM loss

$$L(W) = 1/N \sum_{i=1}^{N} L_i(f(x_i, W), y_i) \quad + \lambda(W)$$

Data loss: "badness" of the model that match the training data

Regularization: Simple model for fitting test data



Tree	**5.5**	3.2	1.3
Dog	1.2	**-2.1**	3.1
Mug	0.5	0.6	**2.1**
Losses	L_1=0	L_2=10	L_3=2.2

L(W) =(0+10+2.2)/3 =4.1

L_1=max(0,1.2-5.5+1)+max(0,0.5-5.5+1)
=0+0
=0

L_2=max(0,3.2--2.1+1)+max(0,0.6--2.1+1)
=6.3+3.7
=10

L_3=max(0,1.3-2.1+1)+max(0,3.1-2.1+1)
=0.2+2.0
=2.2

Multiclass SVM Loss
Hinge loss

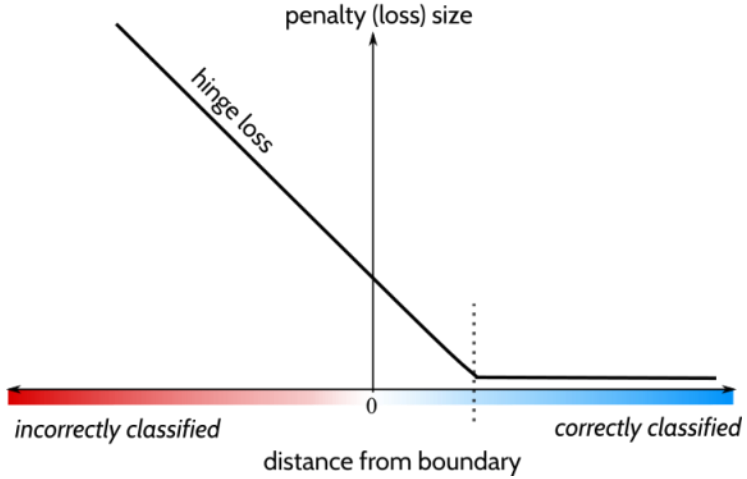


Image source: https://towardsdatascience.com/a-definitive-explanation-to-hinge-loss-for-support-vector-machines-ab6d8d3178f1

$$L_i = \sum_{j \neq y_i} \begin{cases} 0 & \text{if } s_{y_i} \geq s_j + 1 \\ s_j - s_{y_i} + 1 & \text{otherwise} \end{cases}$$
$$= \sum_{j \neq y_i} \max(0, s_j - s_{y_i} + 1)$$

Loss Functions and Regularization

Softmax Classifier (Multinomial Logistic Regression)

$$P(Y = k|X = x_i) = \frac{e^s k}{\sum_j e^s j}$$

s=f(x_i;W)

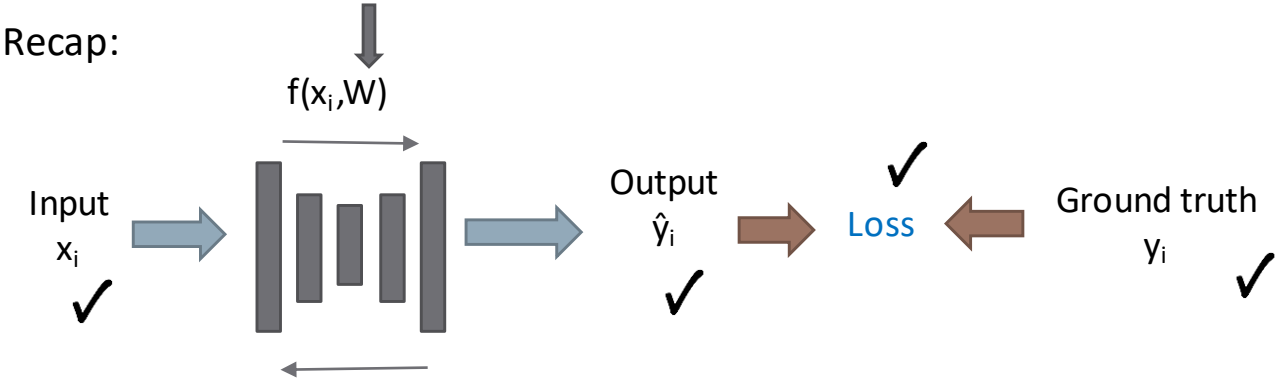
The process where we find the "best" W is optimization

Maximize the log likelihood

L_i= - log P(Y=y_i | X=x_i)

$L_i = -\log\left(\frac{e^s k}{\sum_j e^s j}\right)$

Recap:



	score	e^score	probabilities	-log(probabilities)
Tree	5.5	244.7	0.98	-log(0.98) = -0.008
Dog	1.2	3.3	0.01	-2
Mug	0.5	1.7	0.01	-2

Optimization

- Stochastic Gradient Descent (SGD)

$$L(W) = \frac{1}{N}\sum_{i=1}^{N} L_i(x_i, yi, W) + \lambda R(W)$$

$$\nabla_W L(W) = \frac{1}{N}\sum_{i=1}^{N} \nabla_W L_i(x_i, yi, W) + \lambda \nabla_W R(W)$$

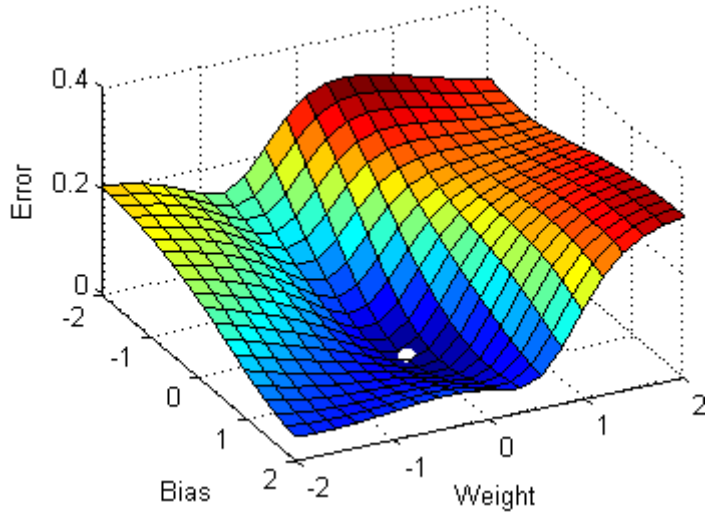


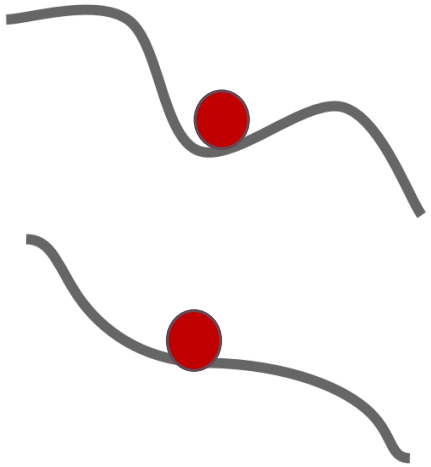
Image source: https://medium.com/@hakobavjyan/stochastic-gradient-descent-sgd-10ce70fea389

Problems with SGD

- Local minima and saddle point
- Gradient get “stuck” in local minima
- Gradient is zero in these saddle points and got “stuck”

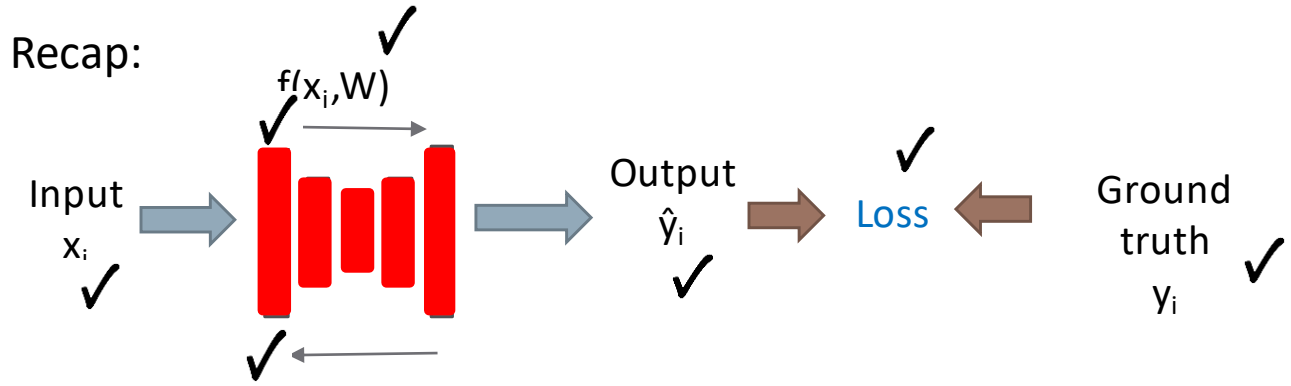
Solutions

- SGD + Momentum
- AdaGrad
- Adam – good default choice



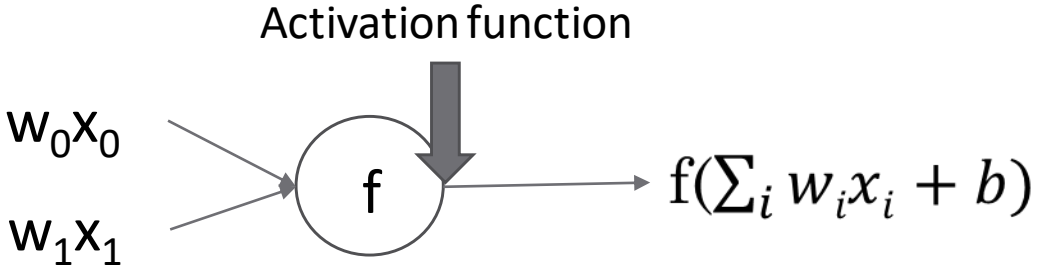
https://www.youtube.com/watch?v=_JB0AO7QxSA

Network Layers



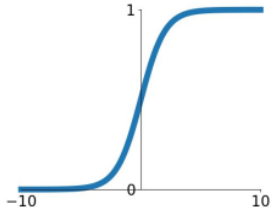
For an additional back propagation slide, please refer to Appendix A

Activation Functions



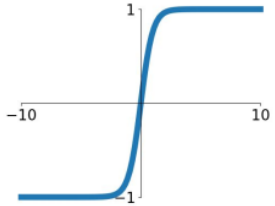
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



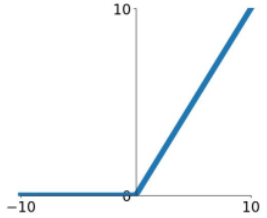
tanh

$$\tanh(x)$$



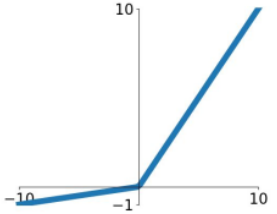
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

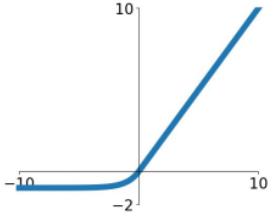


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

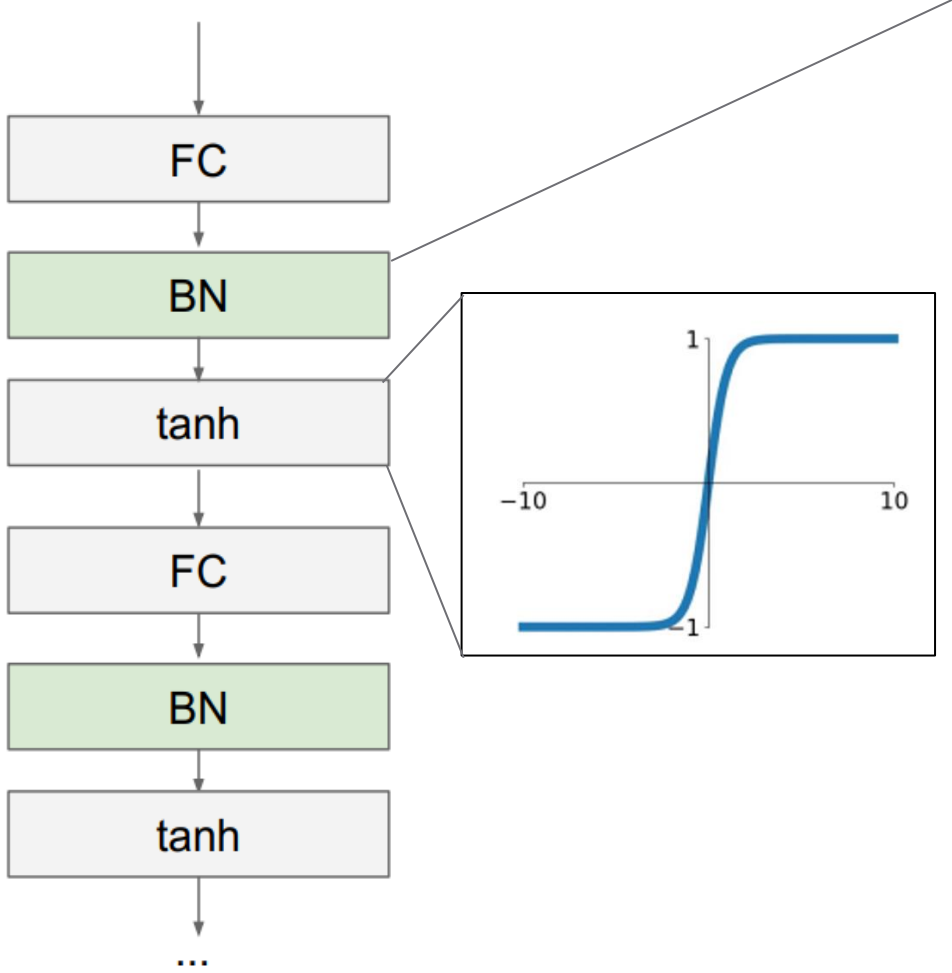
ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



To learn more: https://www.youtube.com/watch?v=wEoyxE0GP2M

Batch Normalization



Input: Values of x over a mini-batch: $\mathcal{B} = \{x_{1...m}\}$;
Parameters to be learned: γ, β

Output: $\{y_i = \mathrm{BN}_{\gamma,\beta}(x_i)\}$

$$\mu_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \qquad \text{// mini-batch mean}$$

$$\sigma^2_{\mathcal{B}} \leftarrow \frac{1}{m}\sum_{i=1}^{m}(x_i - \mu_{\mathcal{B}})^2 \qquad \text{// mini-batch variance}$$

$$\widehat{x}_i \leftarrow \frac{x_i - \mu_{\mathcal{B}}}{\sqrt{\sigma^2_{\mathcal{B}} + \epsilon}} \qquad \text{// normalize}$$

$$y_i \leftarrow \gamma\widehat{x}_i + \beta \equiv \mathrm{BN}_{\gamma,\beta}(x_i) \qquad \text{// scale and shift}$$

Algorithm 1: Batch Normalizing Transform, applied to activation x over a mini-batch.

Supplementary video: https://www.youtube.com/watch?v=wEoyxE0GP2M

Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. ICML'15: Proceedings of the 32nd International Conference on International Conference on Machine Learning, 2015, 448-456. https://arxiv.org/abs/1502.03167

Convolution Layer

0	0	0	0	0	0	...
0	156	155	156	158	158	...
0	153	154	157	159	159	...
0	149	151	155	158	159	...
0	146	146	149	153	158	...
0	145	143	143	148	158	...
...	...	...	...	...	...	...

Input Channel #1 (Red)

0	0	0	0	0	0	...
0	167	166	167	169	169	...
0	164	165	168	170	170	...
0	160	162	166	169	170	...
0	156	156	159	163	168	...
0	155	153	153	158	168	...
...	...	...	...	...	...	...

Input Channel #2 (Green)

0	0	0	0	0	0	...
0	163	162	163	165	165	...
0	160	161	164	166	166	...
0	156	158	162	165	166	...
0	155	155	158	162	167	...
0	154	152	152	157	167	...
...	...	...	...	...	...	...

Input Channel #3 (Blue)

-1	-1	1
0	1	-1
0	1	1

Kernel Channel #1

1	0	0
1	-1	-1
1	0	-1

Kernel Channel #2

0	1	1
0	1	0
1	-1	1

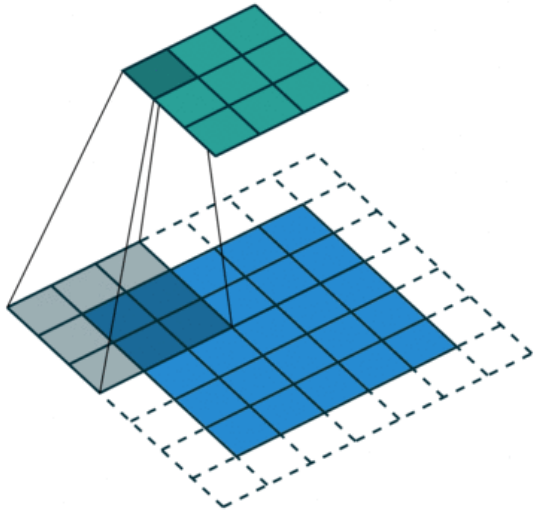
Kernel Channel #3

$308 + (-498) + 164 + 1 = -25$

Bias = 1

Output

-25				...
				...
				...
				...
...	...	...	...	...

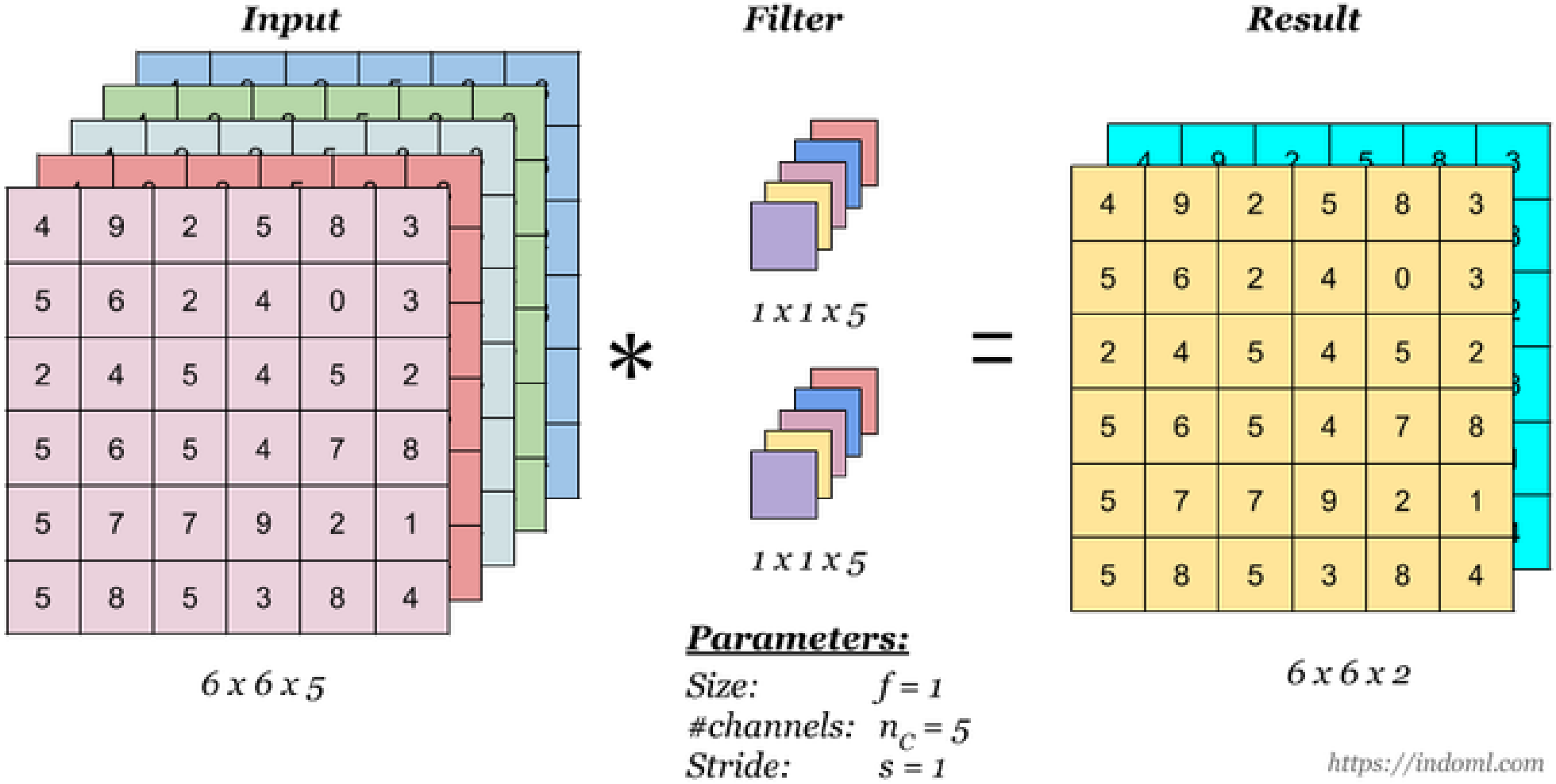


Stride Length = 2
Padding =1

Input size = M x N x 3
Kernel size = 3 x 3 x 3
Output size = M x N x 1

Image credit: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53

1x1 Convolution Layer

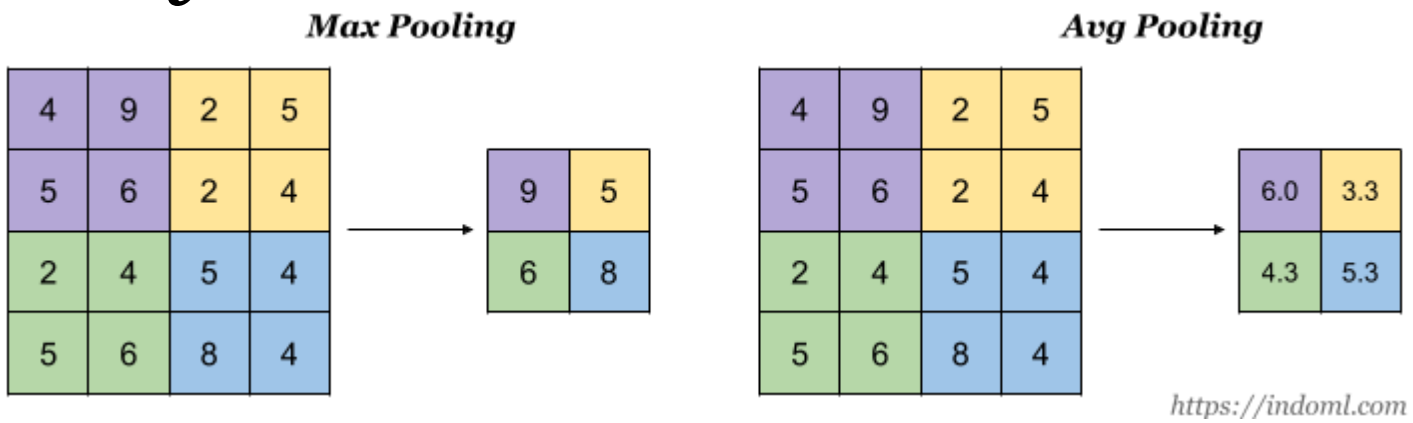


Parameters:

Size: $f = 1$
#channels: $n_C = 5$
Stride: $s = 1$

https://indoml.com

Pooling Layer



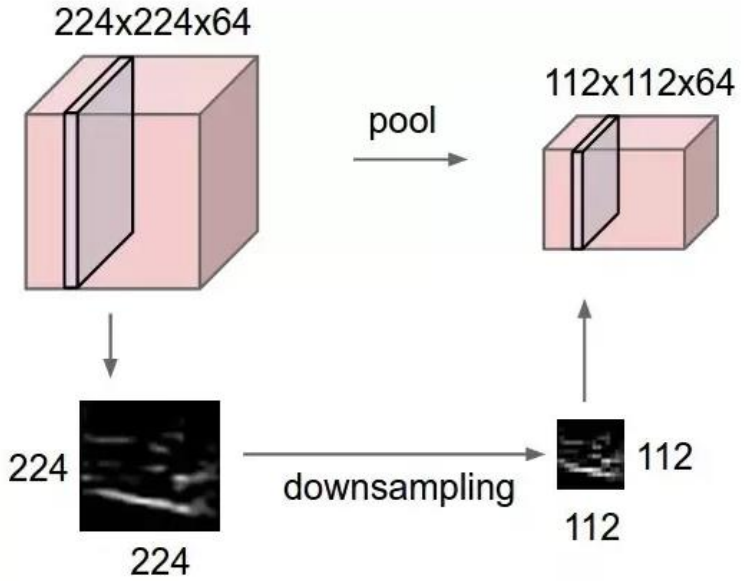
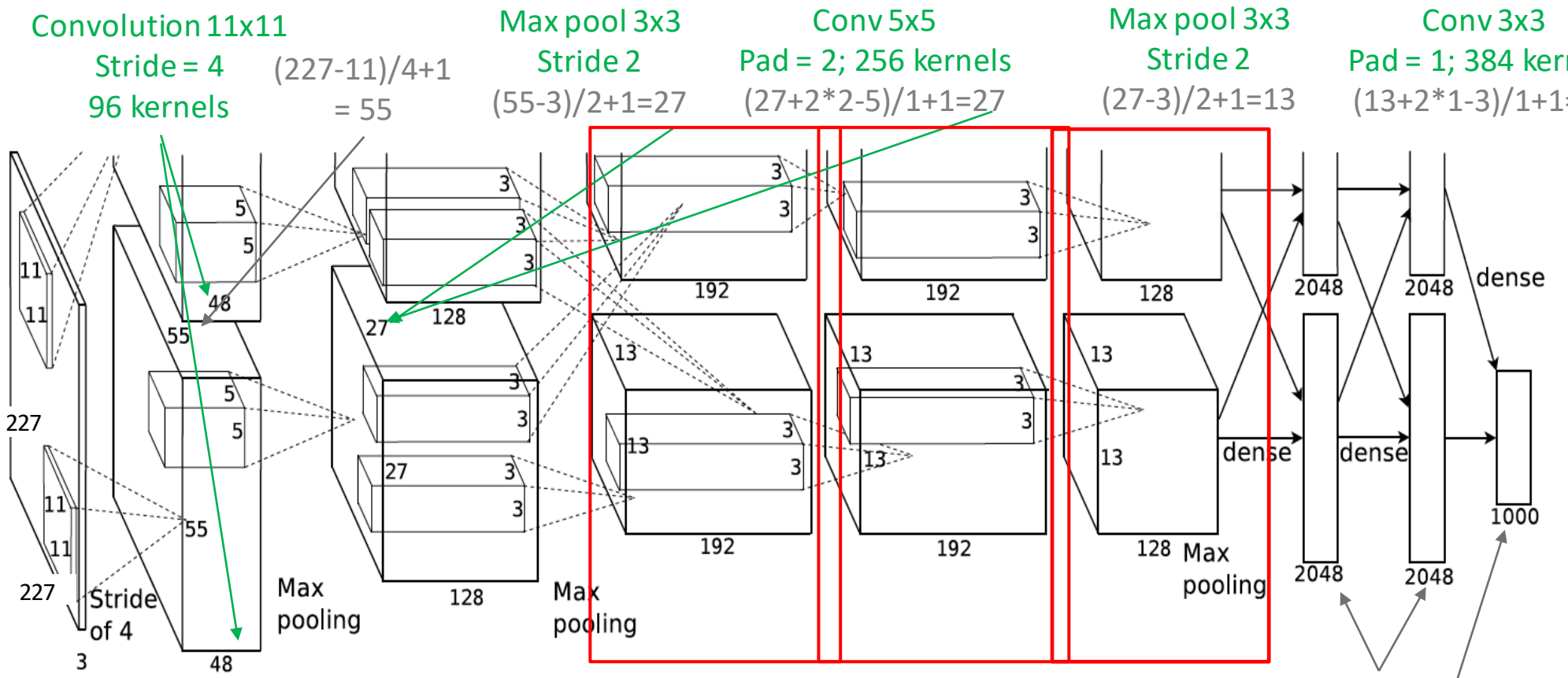


Image credit: https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d
https://computersciencewiki.org/index.php/Max-pooling_/_Pooling

Dimension Exercise



A Krizhevsky, I Sutskever, GE Hinton, Imagenet classification with deep convolutional neural networks, Advances in neural information processing systems 25, 1097-1105
https://papers.nips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

Putting Altogether - Training

- Choosing Activation Function
- Normalize the data
- Weight Initialization (Xavier init)
- Standardize image input, usually square
- Batch normalization
- Batch, Iteration, Epoch
- Checking Loss curve
- Hyperparameters
  - Learning rate
  - Regularization (Strength)

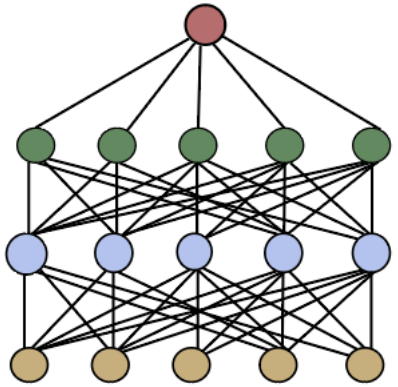
Instead of passing the entire dataset into the network, divided into number of batches, the number of batches is the number of iterations for one epoch

Regularization: L1, L2, Dropout, Drop Layers

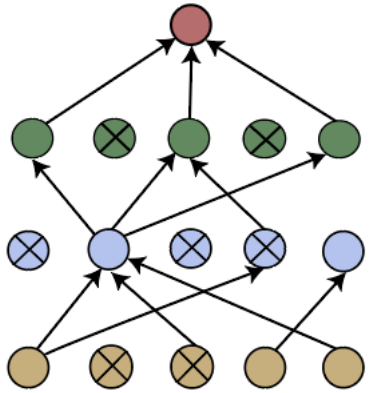
$$L = \frac{1}{N}\sum_{i=1}^{N}\sum_{j\neq y_i}\max(0, f(x_i; W)_j - f(x_i; W)_{y_i} + 1) + \boxed{\lambda R(W)}$$

L1: $R(W) = \sum_k \sum_l |W_{k,l}|$

L2: $R(W) = \sum_k \sum_l W_{k,l}^2$



Standard Neural Network



After Applying Dropout

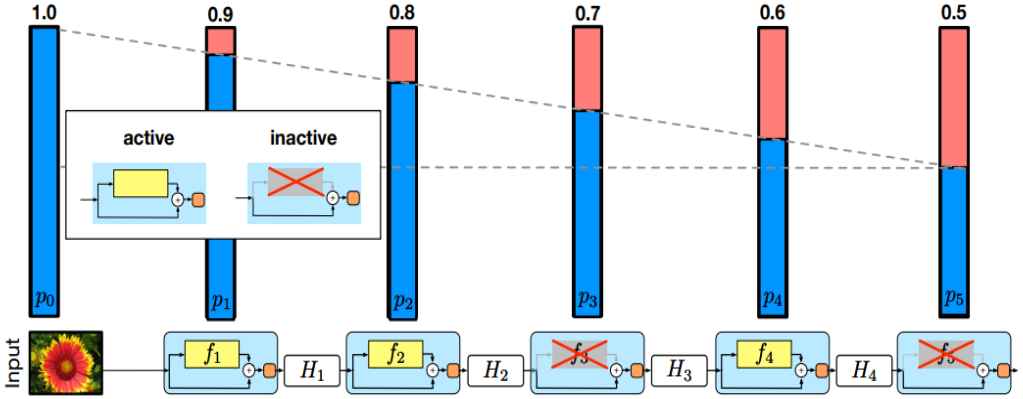
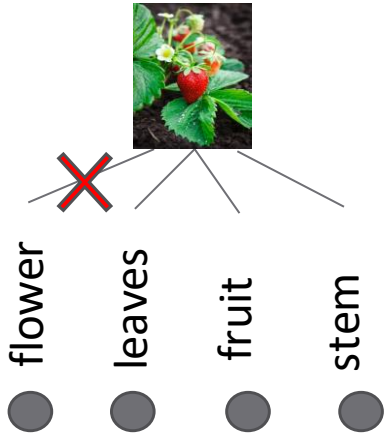


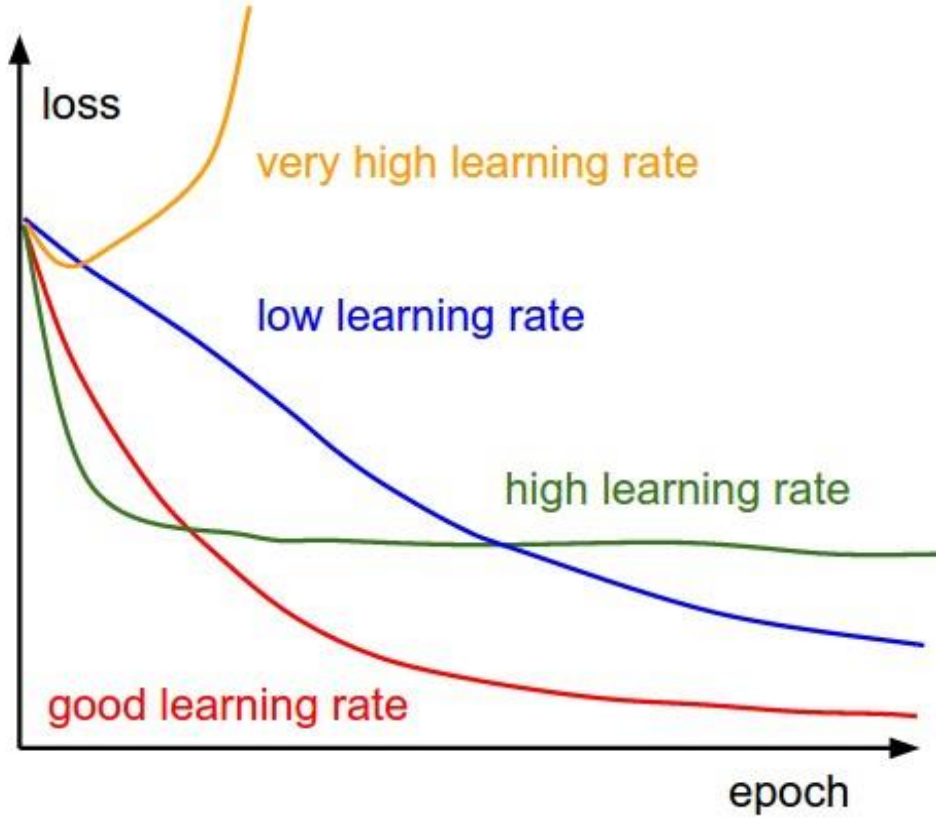
Fig. 2. The linear decay of p_ℓ illustrated on a ResNet with stochastic depth for $p_0 = 1$ and $p_L = 0.5$. Conceptually, we treat the input to the first ResBlock as H_0, which is always active.

G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger. Deep networks with stochastic depth. In ECCV (4), volume 9908 of Lecture Notes in Computer Science, pages 646 - 661. Sprinter, 2016. https://arxiv.org/pdf/1603.09382.pdf



Fore more: https://www.youtube.com/watch?v=_JB0AO7QxSA

Monitoring Loss Curve and Accuracy Curve



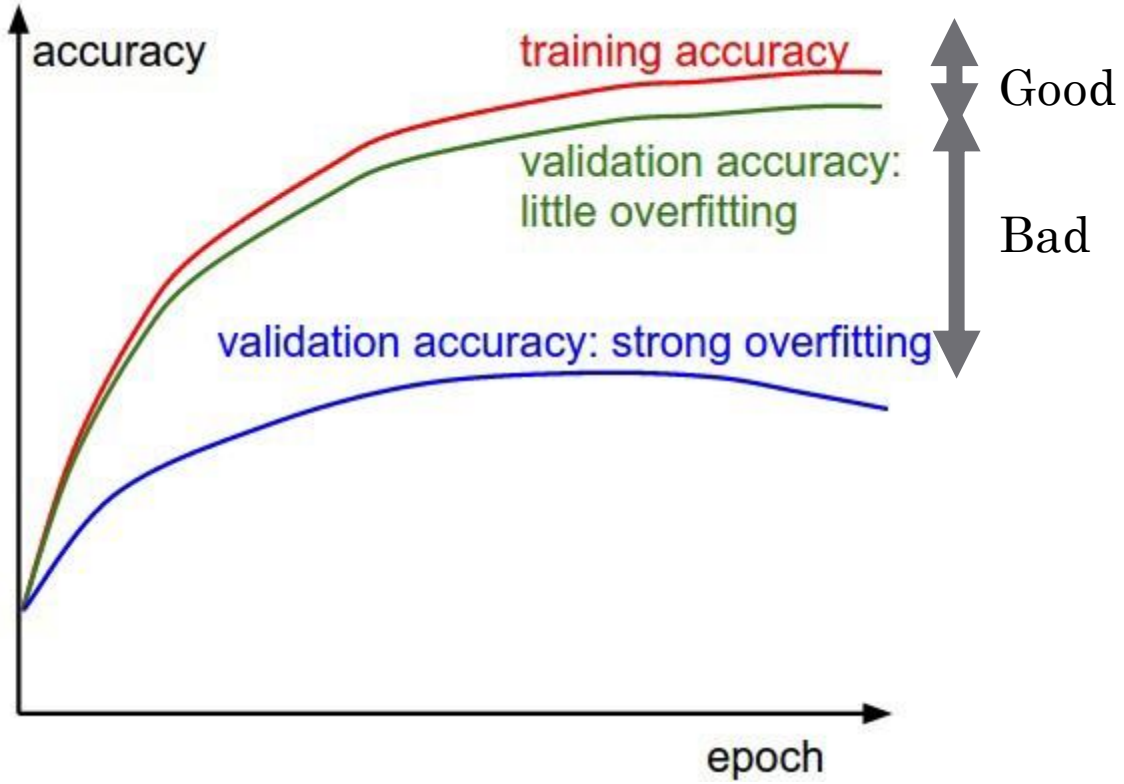
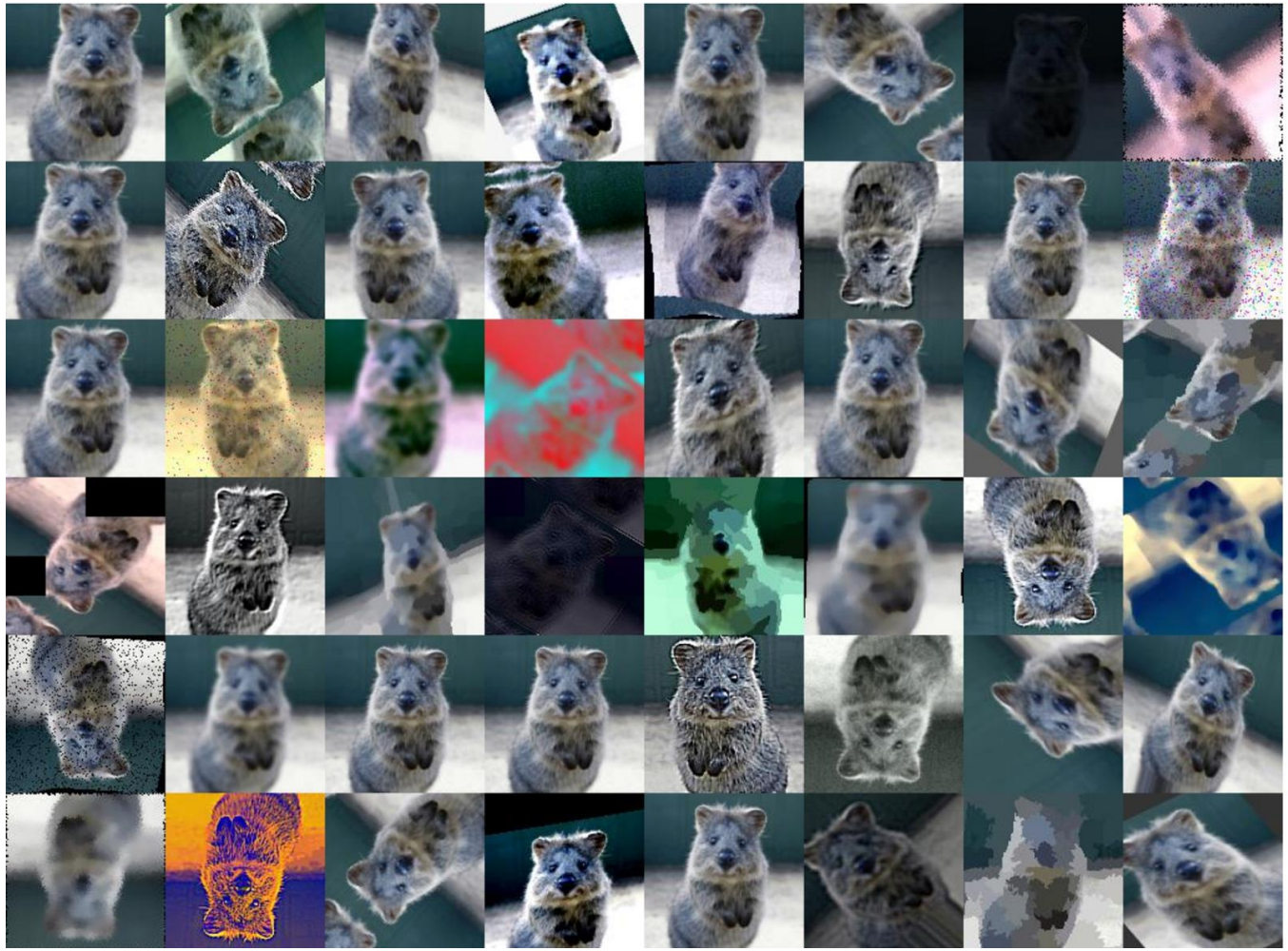


Image source: https://cs231n.github.io/assets/nn3/learningrates.jpeg

Data Augmentation



Random or combinations of
Translation
Rotation
Stretching
Shearing
Padding
Colour jittering
Edge enhancement
……

Image credit: https://laptrinhx.com/introduction-to-dataset-augmentation-and-expansion-3929147356/

Object Detection Papers

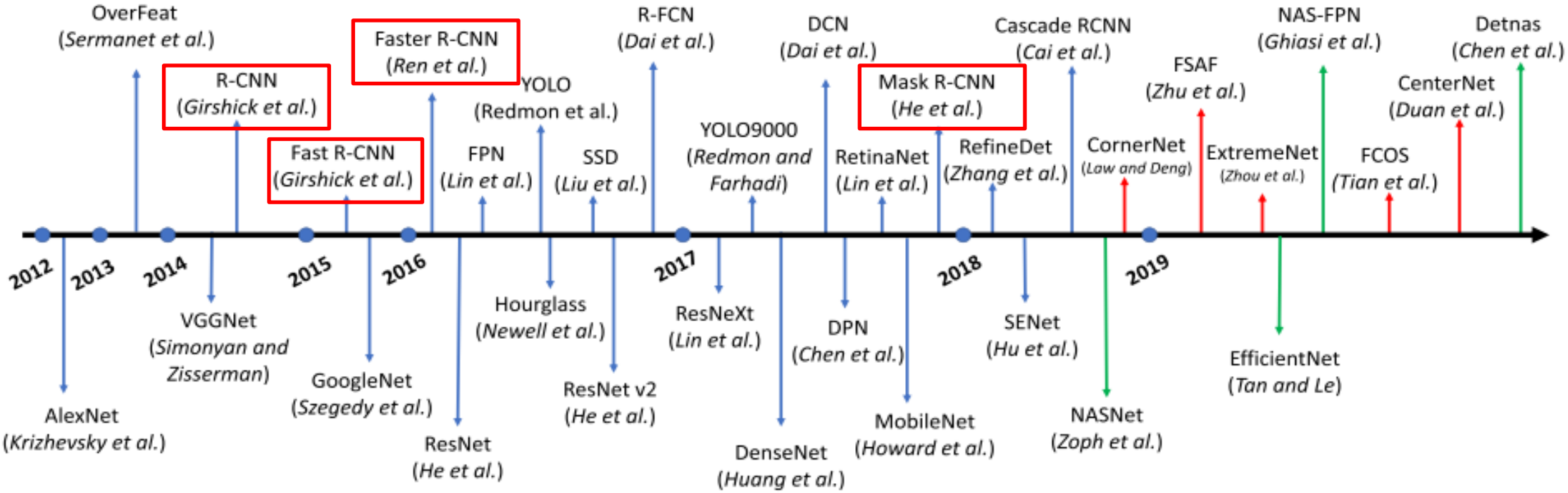


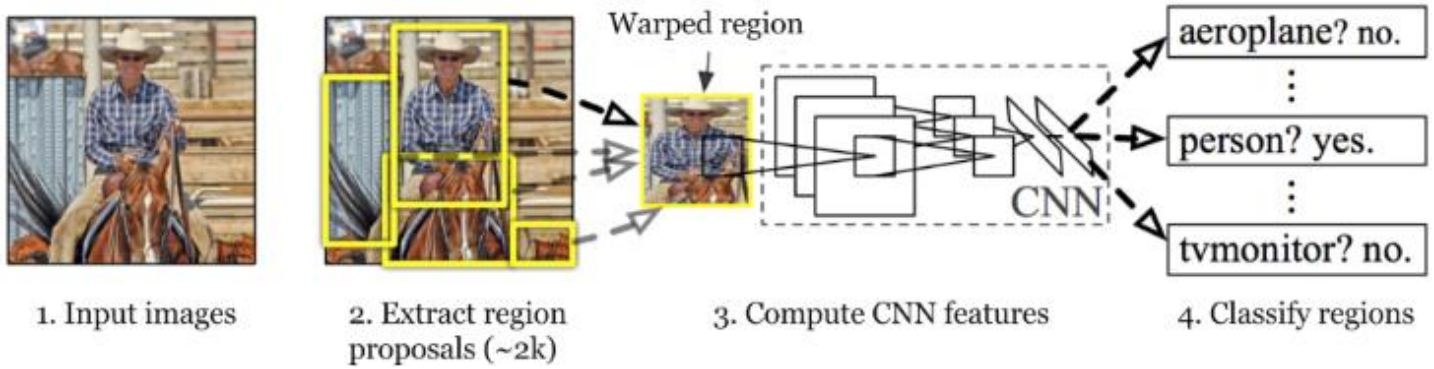
Fig. 2. Major milestone in object detection research based on deep convolution neural networks since 2012. The trend in the last year has been designing object detectors based on anchor-free (in red) and AutoML (in green) techniques, which are potentially two important research directions in the future. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

X. Wu, D. Sahoo and S.C.H. Hoi Recent advances in deep learning for object detection, Neurocomputing 396 (2020) 39–64, https://doi.org/10.1016/j.neucom.2020.01.085.

R-CNN Overview

Region-based Convolutional Neural Networks

Overview of the Process



1. First, using selective search, it identifies a manageable number of bounding-box object region candidates (Region of Interest = RoI)
2. Extracts CNN features from each region independently for classification

Girshick, R.B., Donahue, J., Darrell, T., & Malik, J. (2014). Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 580-587. https://arxiv.org/pdf/1311.2524.pdf

R-CNN Workflow

- Model Workflow
  1. Pre-train a CNN network.
  2. Propose category-independent RoI
  3. Warp Region candidates to have a fixed size as required by CNN.
  4. Continue fine-tuning the CNN on warped regions for K + 1 classes.
  5. Given every image region, one forward propagation through the CNN generates a feature vector.
  6. Train a regression model

Selective search is a common algorithm to provide region proposals that potentially contain objects

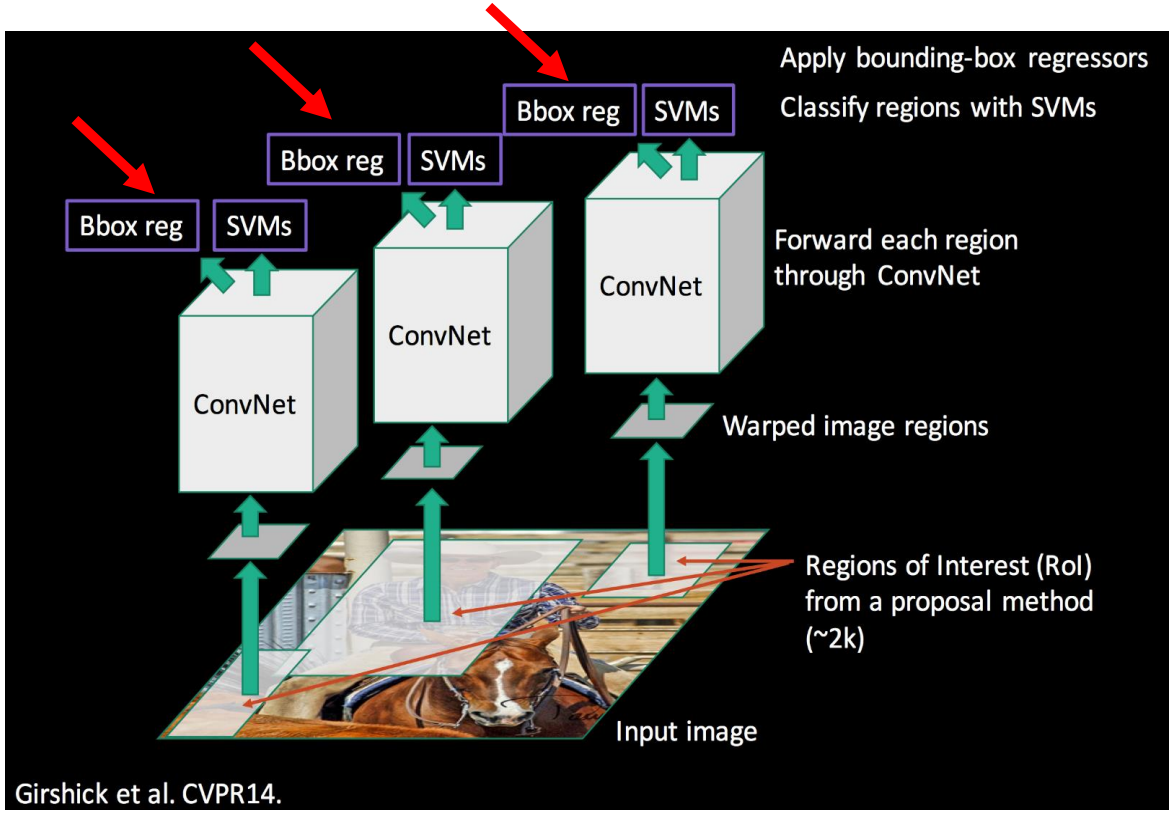
R-CNN Regression and Loss

- Bounding Box Regression
  - $p = (p_x, p_y, p_w, p_h)$: Predicted Bbox coordinates
  - $g = (g_x, g_y, g_w, g_h)$: Ground Truth Bbox coordinates
  - $d_i(p)$: Bbox correction Function
  - t_i : Targets, where

$$
\begin{aligned}
t_x &= (g_x - p_x)/p_w \\
t_y &= (g_y - p_y)/p_h \\
t_w &= \log(g_w/p_w) \\
t_h &= \log(g_h/p_h)
\end{aligned}
$$

  - Loss Function:

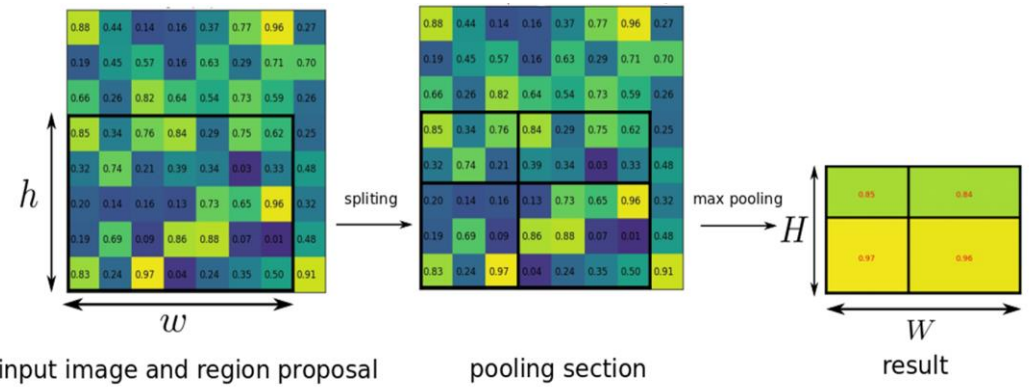
$$\mathcal{L}_{\text{reg}} = \sum_{i \in \{x,y,w,h\}} (t_i - d_i(\mathbf{p}))^2 + \lambda \|\mathbf{w}\|^2$$



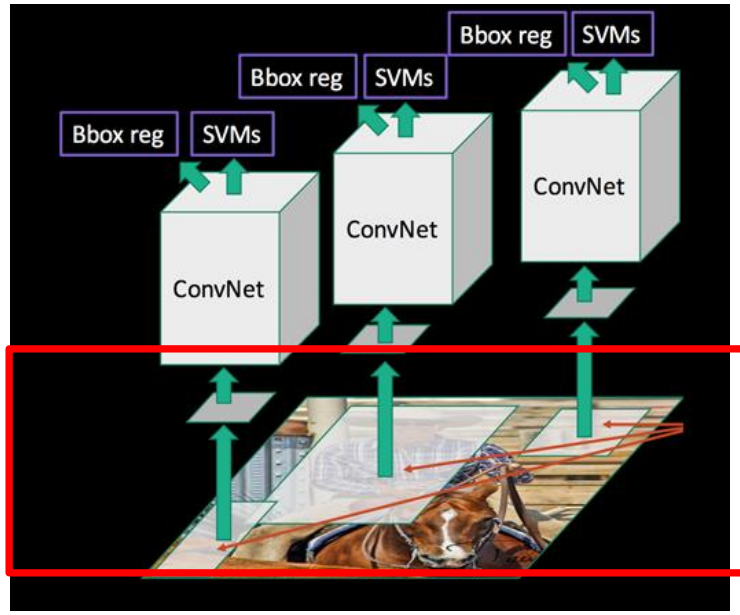
Girschick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014

Fast R-CNN Overview

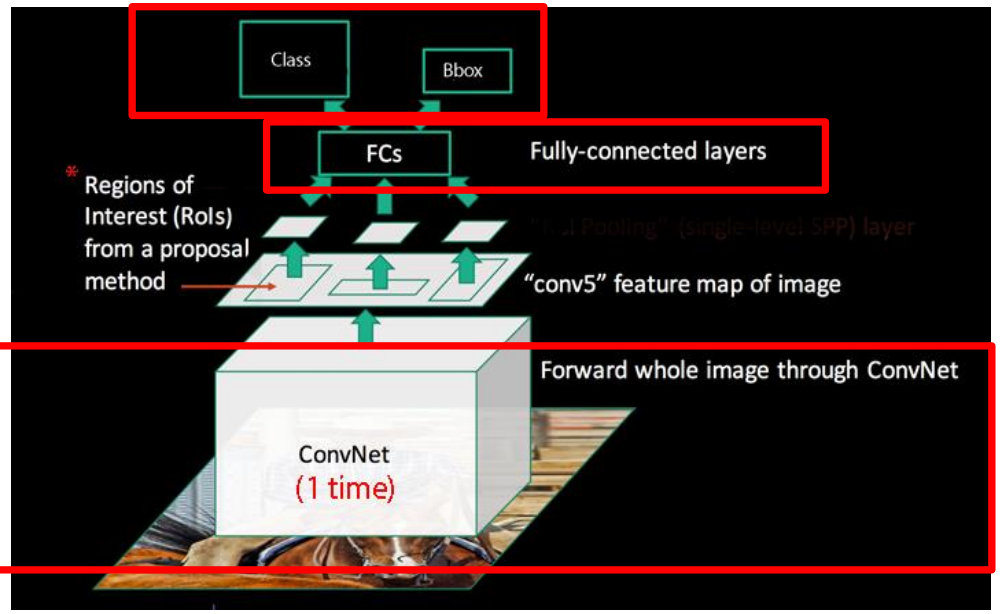
- How can we do it faster?
  - Unifying models that used to work independently



RCNN



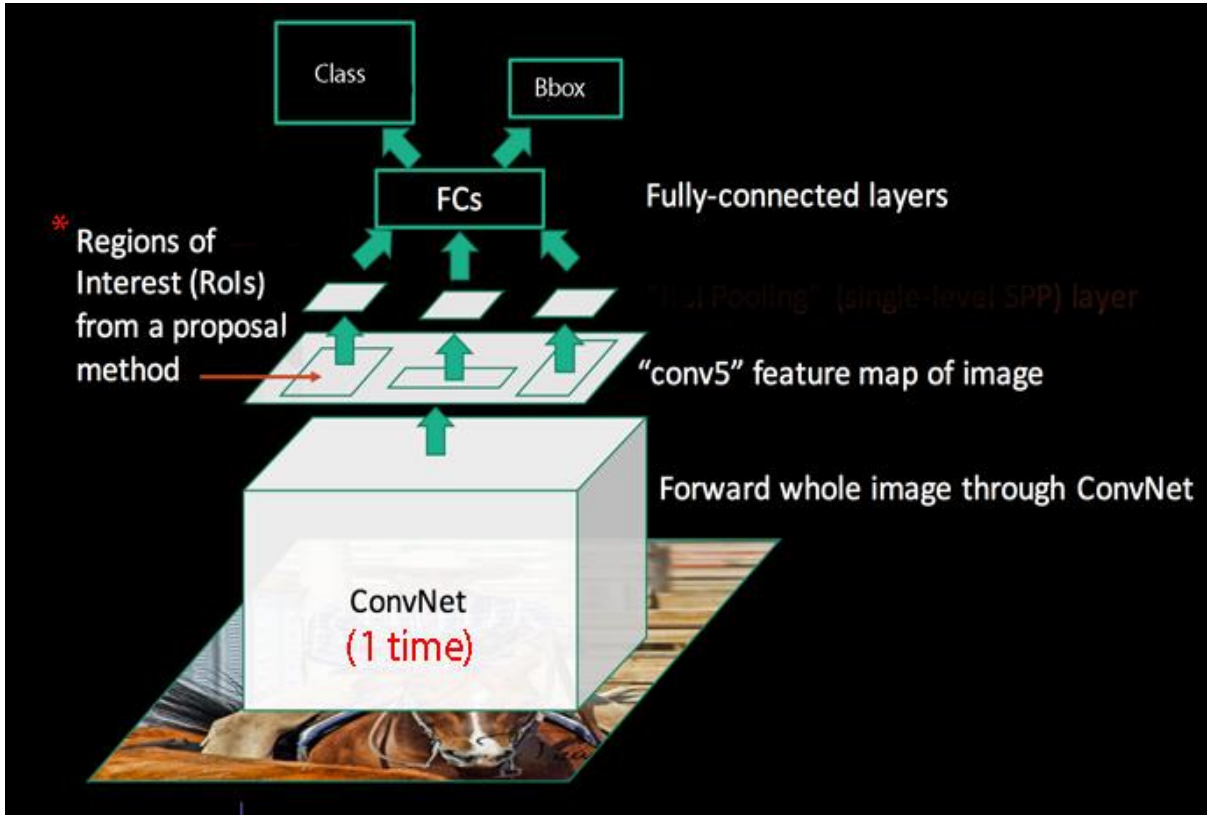
Fast RCNN



R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1440-1448, doi: 10.1109/ICCV.2015.169.
Image source: https://deepsense.ai/region-of-interest-pooling-explained/

Fast R-CNN Workflow

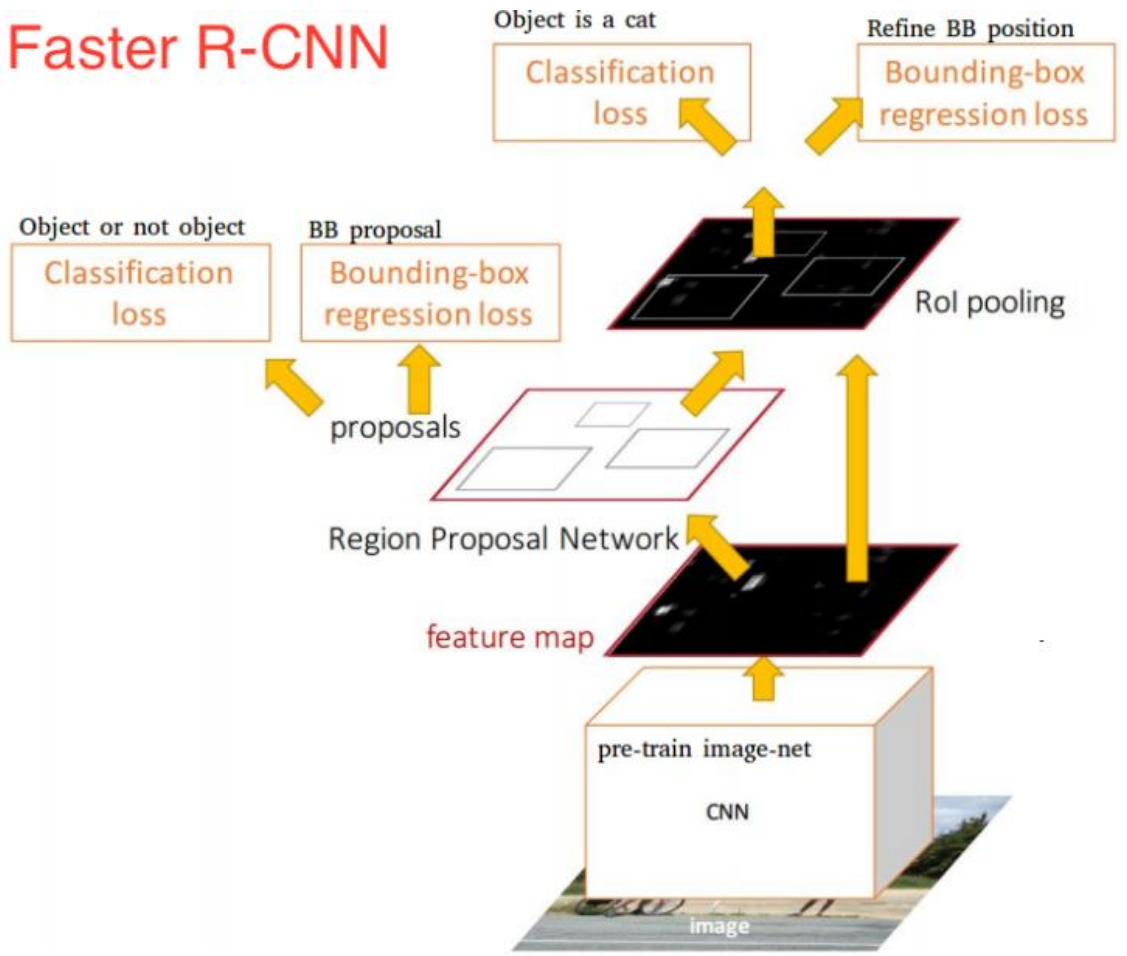
1. Pre-train a CNN on image classification tasks
2. Propose regions by selective search (~2k candidates per image)
3. Alter the pre-trained CNN:
   - Replace pooling layer RoI pooling layer
   - Replace fully connected layer and softmax layer (K classes) with K + 1 classes
4. Branch into two output layers:
   - A softmax estimator of K + 1 classes
   - A bounding-box regression model



For additional loss function slide (Fast RCNN), please refer to Appendix B

Faster R-CNN Overview

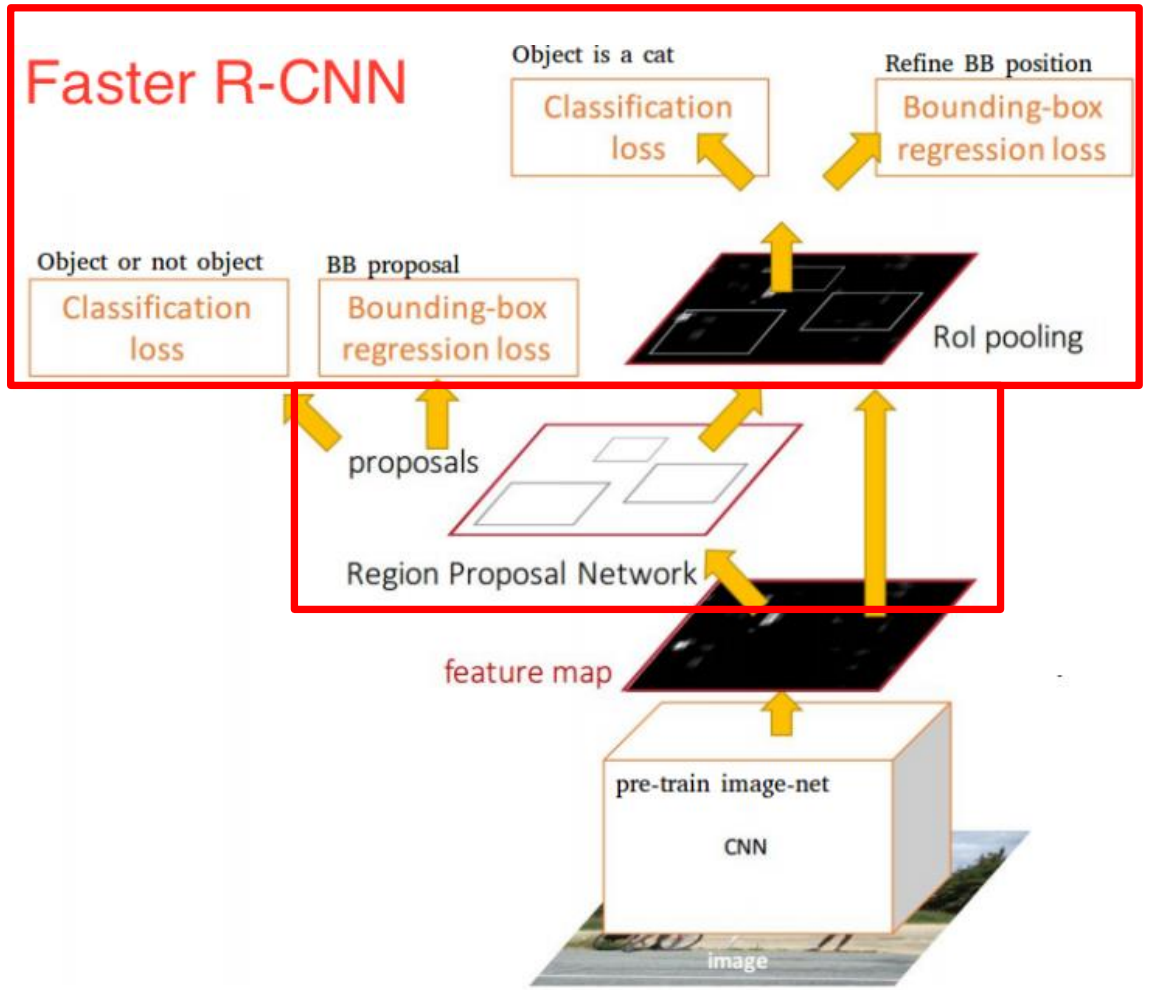
- How can we do it even faster?
  - Construct a single, unified model composed of Region Proposal Network (RPN)



S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

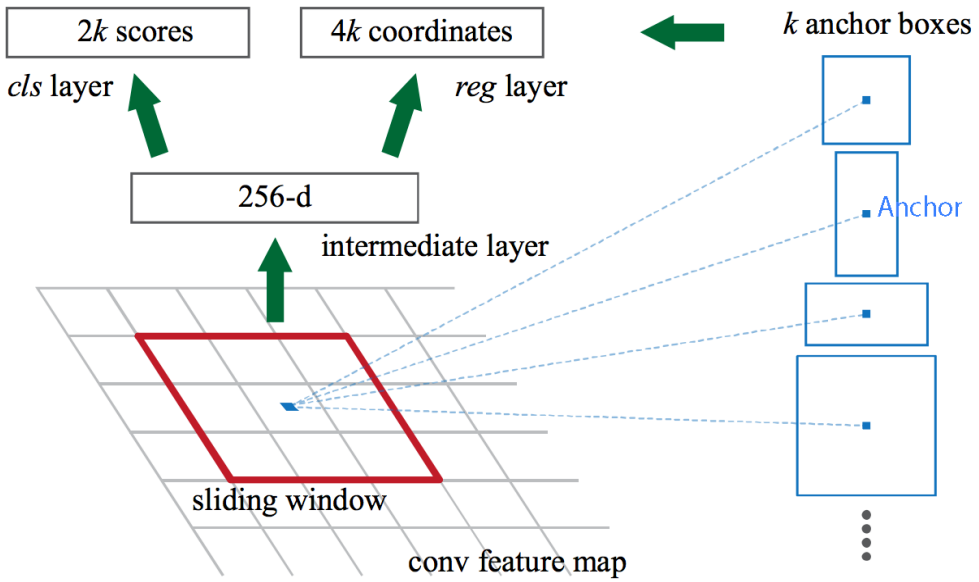
Faster R-CNN Workflow

- Model Workflow
  1. Pre-train a CNN on image classification tasks
  2. Fine-tune the RPN end-to-end for the region proposal task
     - initialized by the pre-train image classifier
  3. Train a Fast R-CNN object detection model
     - using the proposals generated by the current RPN
  4. Jointly train 4 losses
     - Object or not object
     - Regress box coordinate
     - Classification Score
     - 2nd round regress box coordinate



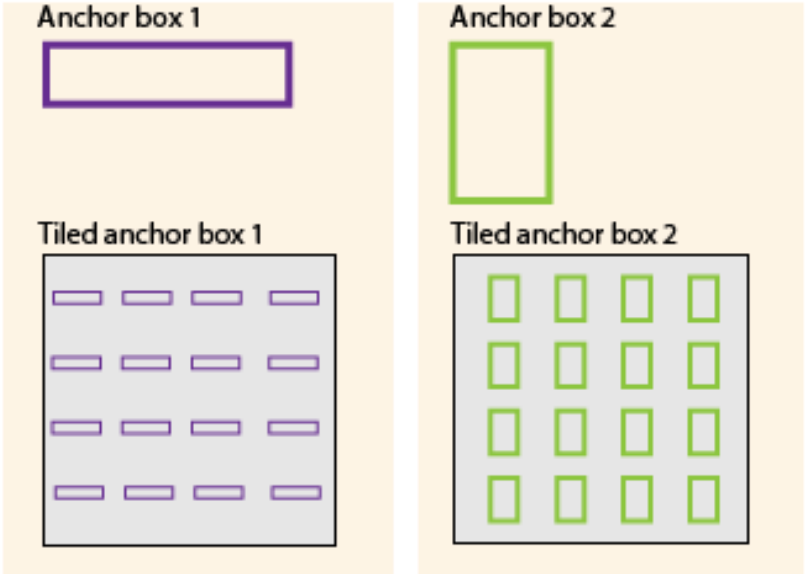
Faster R-CNN – RPN

- Regional Proposal Network
  - Input: feature map from CNN (**NOT IMAGE**)
  - Slide a small n x n window over the feature map of the entire image.
  - At the center of each sliding window, we predict multiple regions of various scales and ratios simultaneously
  - Anchor: sliding window center, scale, ratio
    - For example, 3 scales x 3 ratios => 9 anchors boxes



Faster R-CNN - Anchor Boxes

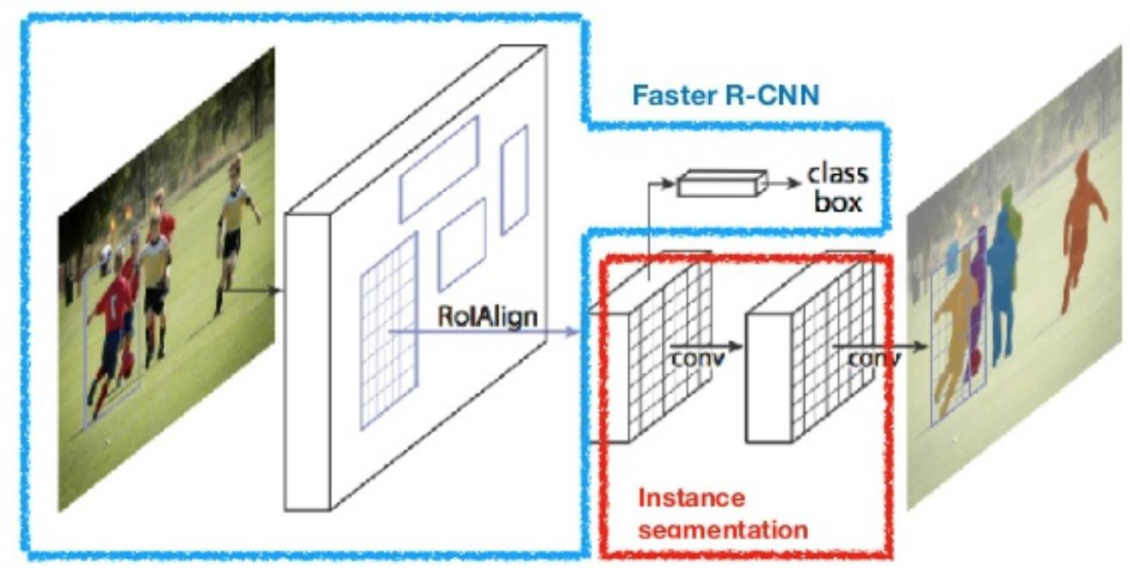
- What is Anchor Box?
  - PREDEFINED bounding boxes of a certain height and width
  - Based on object sizes in the training dataset



For additional loss function slide (Faster RCNN), please refer to Appendix C

Mask R-CNN

- What's the difference?
  - The framework is built on top of Faster R-CNN
  - Decoupling the classification and the pixel-level mask prediction tasks

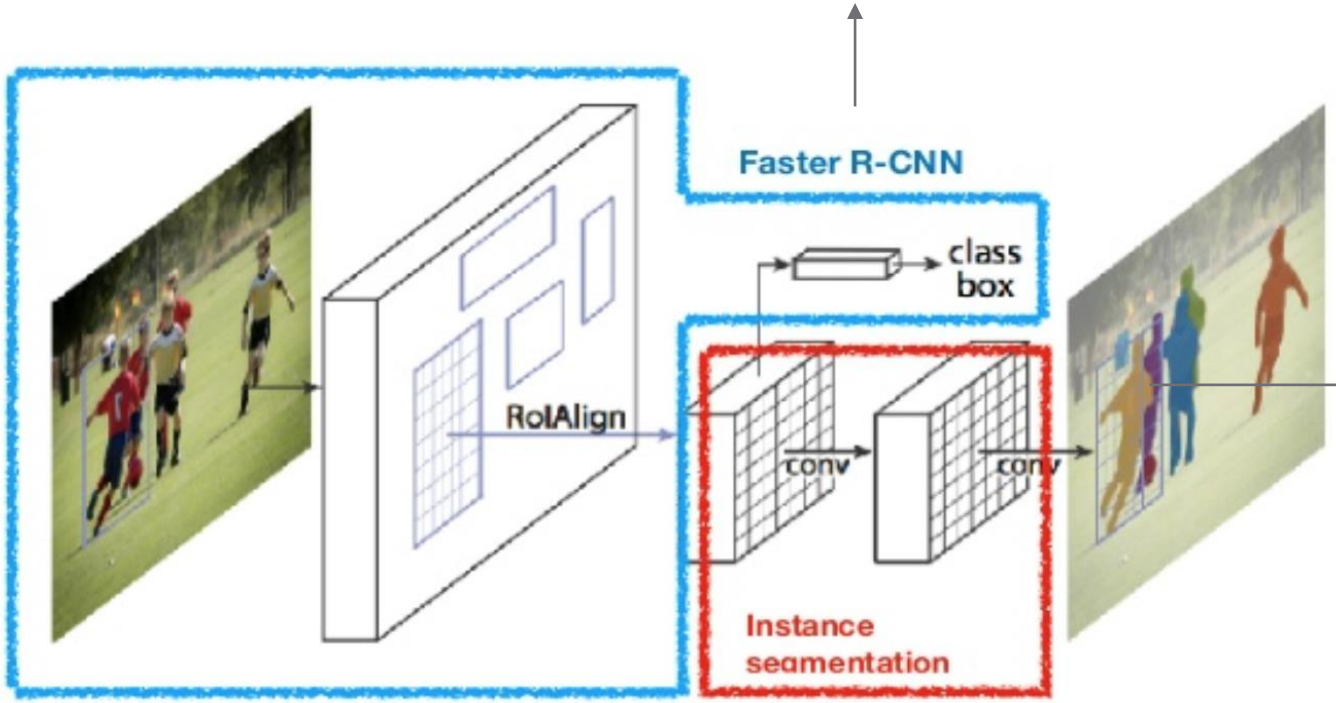


Instance Segmentation

K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.
https://arxiv.org/abs/1703.06870

Mask R-CNN

Just like Faster R-CNN
- Object Classification
- Bounding Box Regressor



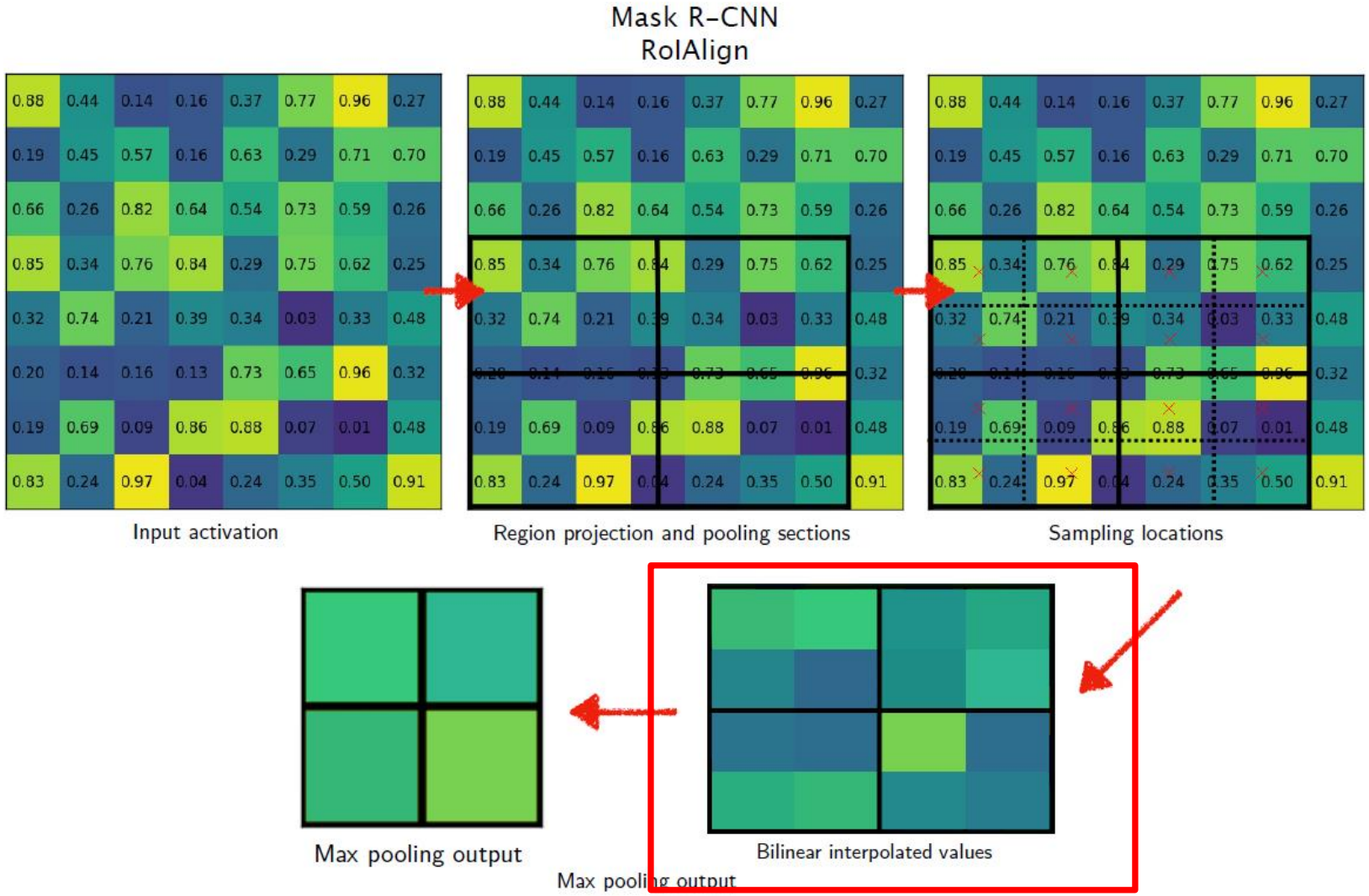
With RoIAlign
- Average Pool

Therefore for Each RoI
- Multi-class Classification
- Bounding Box Regressor

For additional loss function slide (Mask RCNN), please refer to Appendix D

RoIAlign

- What is RoIAlign?
  - Fix location misalignment caused by quantization.
  - Removes the hash quantization



For more: https://medium.com/analytics-vidhya/review-mask-r-cnn-instance-segmentation-human-pose-estimation-61080a93bf4

Summary

Basics:

- Loss Functions and Regularization
- Optimization
- Convolution Layer
- Pooling Layer
- Activation
- Batch Normalization
- Fully Connected Layers

Training:

- Reading accuracy curve and loss curve
- Data augmentation

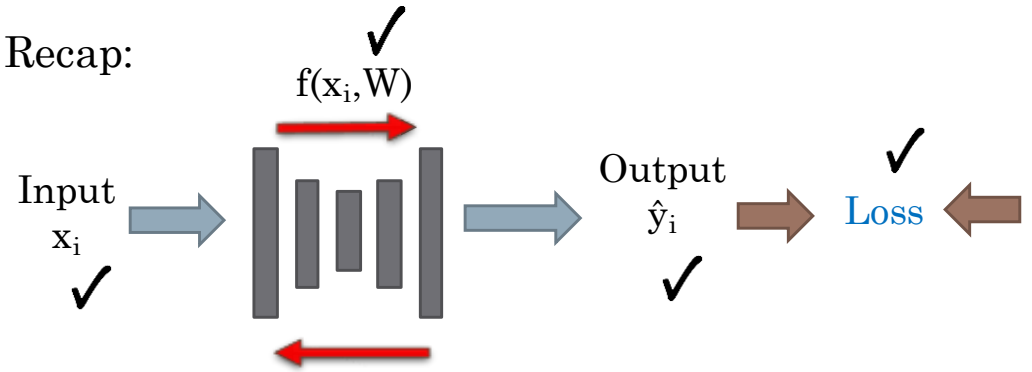
Detection Networks:

- RCNN
- Fast RCNN
- Faster RCNN
- Mask RCNN

Thank you

Appendix A: Back Propagation

Recap:



Want: $\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}$

Example

f(x,y,z)=(x+y)z

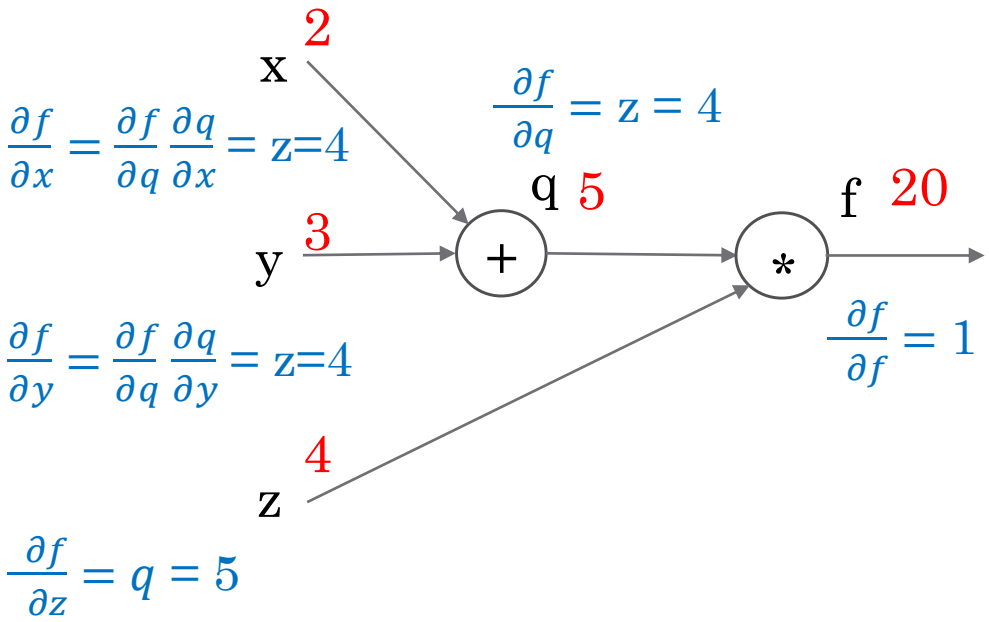
Given x = 2, y = 3, z =4

$q = x + y \qquad \frac{\partial q}{\partial x} = 1, \ \frac{\partial q}{\partial y} = 1$

$f = qz \qquad \frac{\partial f}{\partial q} = z, \frac{\partial f}{\partial z} = q$

We also know from chain rule: $\frac{\partial f}{\partial y} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial y}$

We also know from chain rule: $\frac{\partial f}{\partial x} = \frac{\partial f}{\partial q}\frac{\partial q}{\partial x}$



Full example and explanation: https://www.youtube.com/watch?v=d14TUNcbn1k

Appendix B: Fast-RCNN Loss

- Loss Function
  - u : True class label (background class: u = 0)
  - p : Discrete probability distribution (per RoI) over K + 1
  - v : True bounding box
  - t^u : Predicted bounding box correction

$$\mathcal{L}(p, u, t^u, v) = \mathcal{L}_{\text{cls}}(p, u) + 1[u \geq 1]\mathcal{L}_{\text{box}}(t^u, v)$$
$$\mathcal{L}_{\text{cls}}(p, u) = -\log p_u$$
$$\mathcal{L}_{\text{box}}(t^u, v) = \sum_{i \in \{x,y,w,h\}} L_1^{\text{smooth}}(t_i^u - v_i)$$

  - Loss Function: $L = L_{cls} + L_{box}$
  - Overall Loss Function:

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

39

Appendix C: Faster-RCNN Loss

- Loss Function
  - p_i : Predicted probability of anchor i being an object
  - p_i^* : Ground truth label (binary) of whether anchor i is an object
  - t_i : Predicted four parameterized coordinates
  - t_i^* : Ground truth coordinates
  - N_{cls} : Normalization term, set to be mini-batch size
  - N_{cls} : Normalization term, set to the number of anchor locations
  - Loss Function: $L = L_{cls} + L_{box}$
  - Overall Loss Function:

$$\mathcal{L}(\{p_i\},\{t_i\}) = \frac{1}{N_{\text{cls}}}\sum_i \mathcal{L}_{\text{cls}}(p_i, p_i^*) + \frac{\lambda}{N_{\text{box}}}\sum_i p_i^* \cdot L_1^{\text{smooth}}(t_i - t_i^*)$$

where

$$\mathcal{L}_{\text{cls}}(p_i, p_i^*) = -p_i^* \log p_i - (1 - p_i^*)\log(1 - p_i)$$

Appendix D: Mask-RCNN Loss

- Loss Function
  - Loss Function: $L = L_{cls} + L_{box} + L_{mask}$
  - where,

$$\mathcal{L}_{\text{mask}} = -\frac{1}{m^2} \sum_{1 \le i,j \le m} \left[y_{ij} \log \hat{y}^k_{ij} + (1 - y_{ij}) \log(1 - \hat{y}^k_{ij}) \right]$$

* Binary cross entropy:

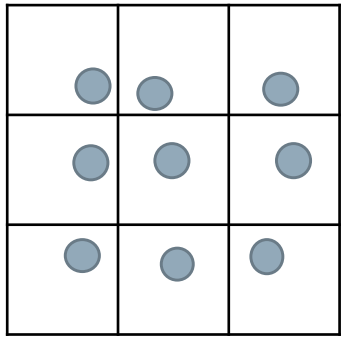
$$L(y, \hat{y}) = -\frac{1}{N} \sum_{i=0}^{N} (y * log(\hat{y}_i) + (1 - y) * log(1 - \hat{y}_i))$$

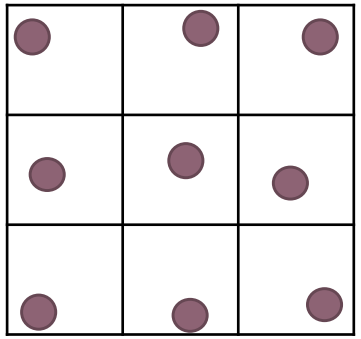
Deep Learning in 3D

Dr. Connie Ko

Challenges in Point Cloud Processing

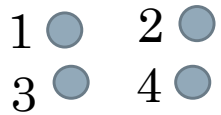
Irregular



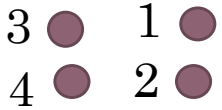


Problem: Two different point clouds end up having same representation

Order-less

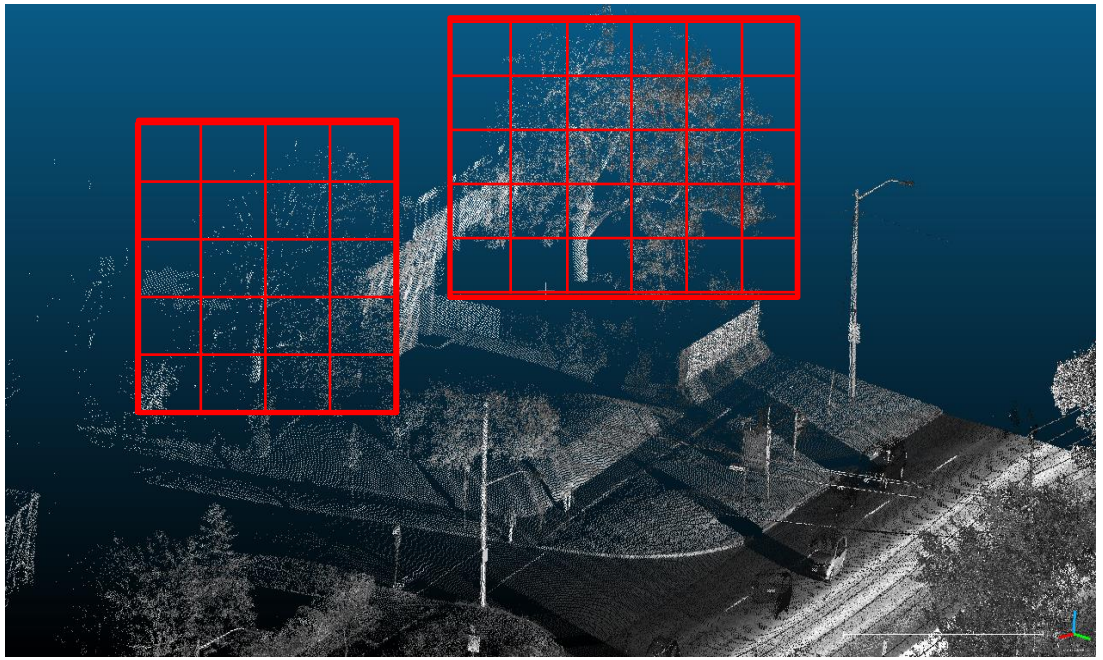


	x	y	z
1	x_1	y_1	z_1
2	x_2	y_2	z_2
3	x_3	y_3	z_3
4	x_4	y_4	z_4



	x	y	z
3	x_1	y_1	z_1
1	x_2	y_2	z_2
4	x_3	y_3	z_3
2	x_4	y_4	z_4

Non-uniform, varying densities



Processing Overview

1. Multi-view
2. Volumetric representation
3. Point based
   a) Point based mlp
   b) Point based conv
4. Graph based
5. RNN based

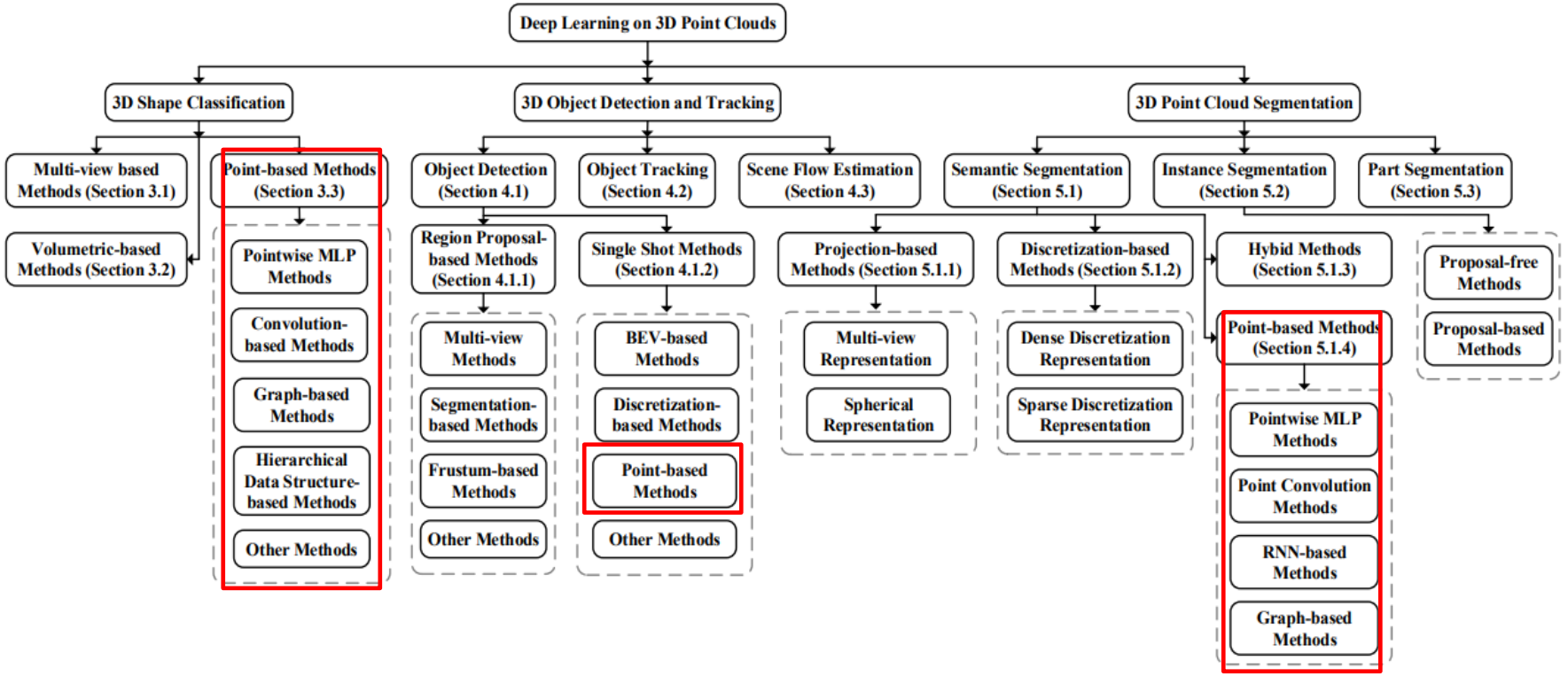
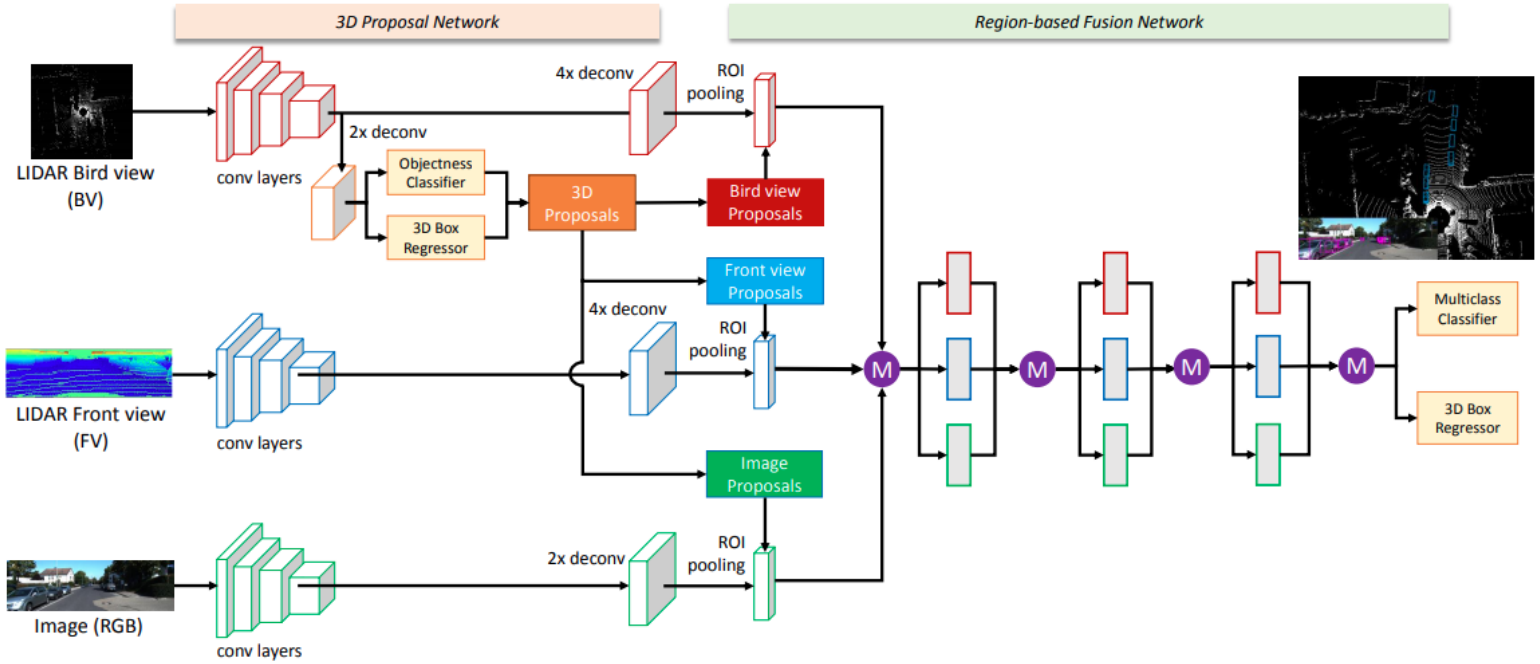


Fig. 1: A taxonomy of deep learning methods for 3D point clouds.

Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun, 2020 "Deep Learning for 3D Point Clouds: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence,doi: 10.1109/TPAMI.2020.3005434.

1. Multi-view

- Basic idea
  - Projection of point clouds to regular structures, e.g. multi-view images, spherical view



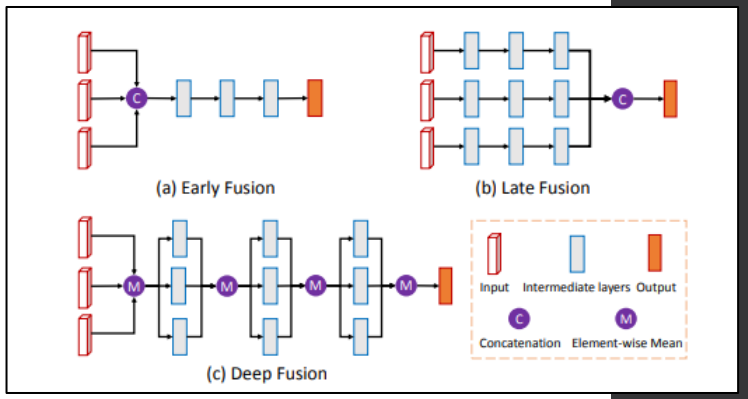


Figure 1: **Multi-View 3D object detection network (MV3D):** The network takes the bird's eye view and front view of LIDAR point cloud as well as an image as input. It first generates 3D object proposals from bird's eye view map and project them to three views. A deep fusion network is used to combine region-wise features obtained via ROI pooling for each view. The fused features are used to jointly predict object class and do oriented 3D box regression.

X. Chen, H. Ma, J. Wan, B. Li and T. Xia, "Multi-view 3D Object Detection Network for Autonomous Driving," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6526-6534, doi: 10.1109/CVPR.2017.691.

1. Multi-view

- Basic idea
  - Projection of point clouds to regular structures, e.g. multi-view images, spherical view

Aggregate View Object Detection (AVOD)

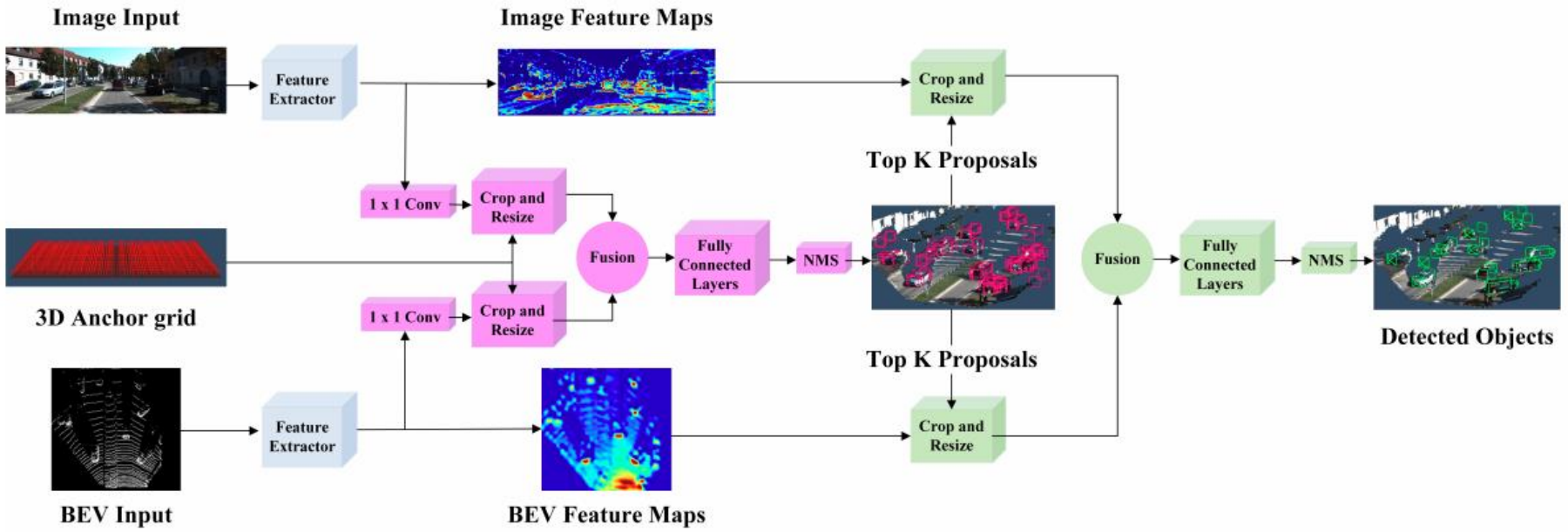


Fig. 2: The proposed method's architectural diagram. The feature extractors are shown in **blue**, the region proposal network in **pink**, and the second stage detection network in **green**.

J. Ku, M. Mozifian, J. Lee, A. Harakeh and S. L. Waslander, "Joint 3D Proposal Generation and Object Detection from View Aggregation," 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1-8, doi: 10.1109/IROS.2018.8594049.

1. Multi-view

- Pros
  - Able to use networks that are built for regular data. Off-the-shelf
  - Efficient and well-established image processing and understanding
  - Use pretrained features
- Cons
  - Information loss from projection
  - Assume all objects are visible by pre-set views
  - “Best” position for camera
  - Rely on camera-LiDAR fusion algorithm

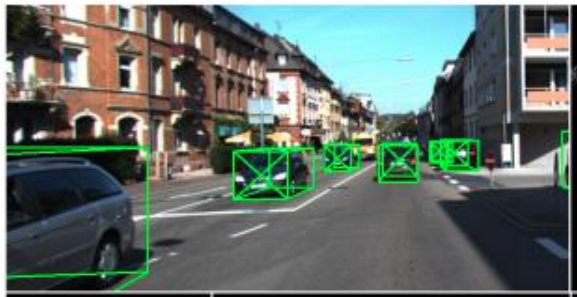


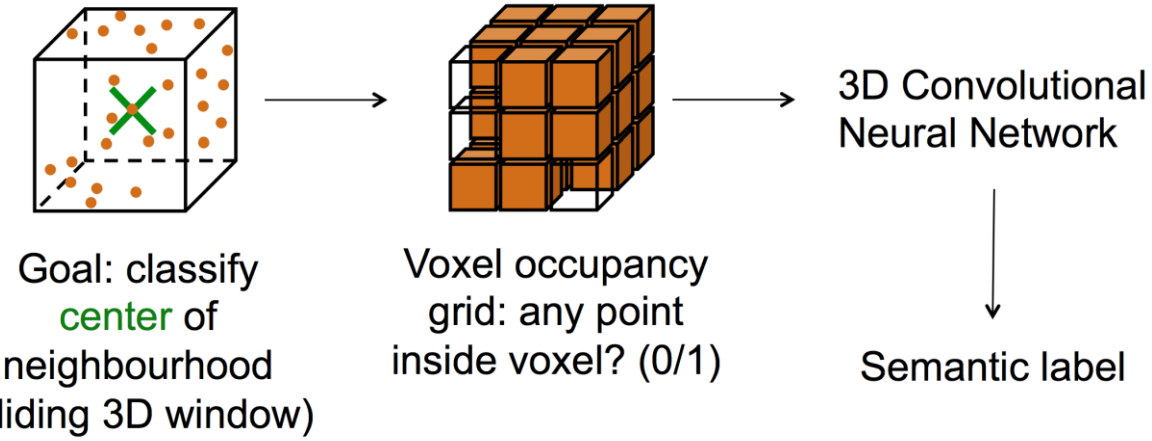
	Image	Point Cloud
Permutation	Ordered	Orderless
Data Structure	Regular	Irregular
Data Type	Discrete	Continuous
Dimension	2D	3D
Coordinates	Projective	Euclidean
Resolution	High	Low

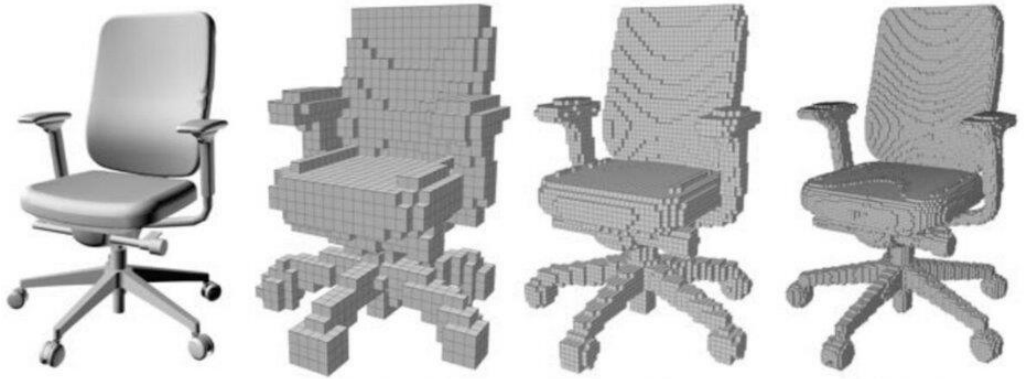
Fig. 1. A comparison between image data and point cloud data.

Y. Cui et al., "Deep Learning for Image and Point Cloud Fusion in Autonomous Driving: A Review," in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2020.3023541.

2. Volumetric Representation

- Basic idea: usually applied on voxelized grid, and then apply 3D CNN on the volume
- Occupancy grid
- Octree
- Dense convolution vs sparse convolution
- Voxel CNN, Spherical CNN



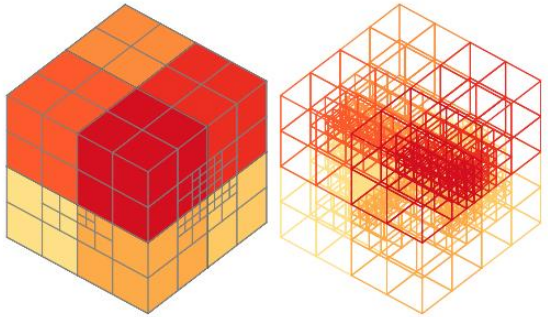


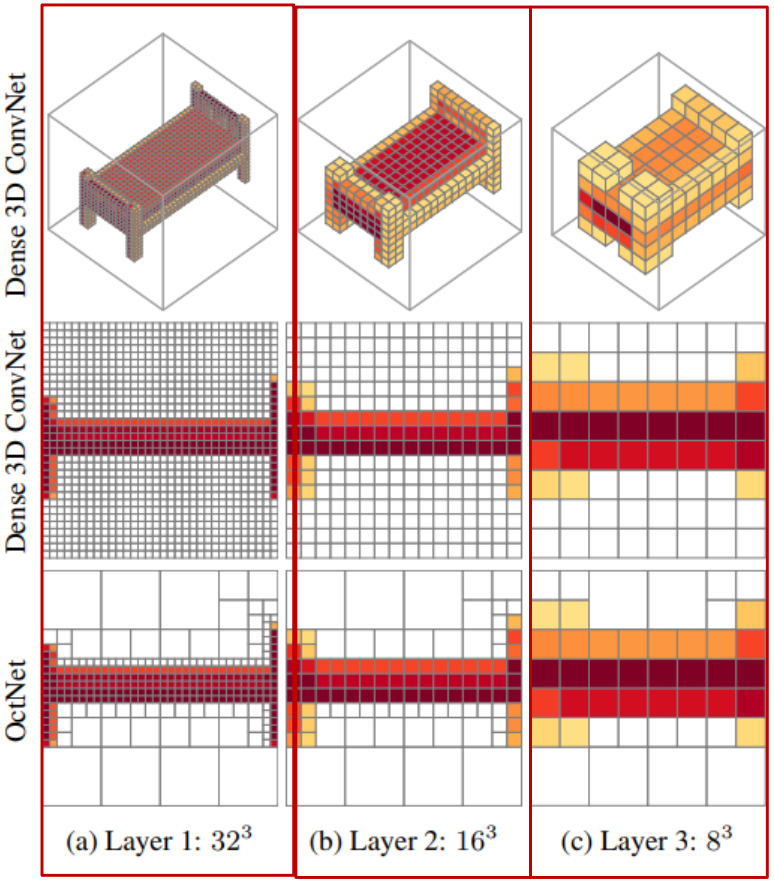
% occupied voxels:	10.41%	5.09%	2.41%
Resolution:	30 x 30 x 30	64 x 64 x 64	128 x 128 x 128

Image source: https://github.com/nsavinov/semantic3dnet
https://www.antoinetlc.com/blog-summary/3d-data-representations

2. Volumetric Representation

- Basic idea: usually applied on voxelized grid, and then apply 3D CNN on the volume
- Occupancy grid
- Octree
- Dense convolution vs sparse convolution
- Voxel CNN, Spherical CNN





1	1	2	3
2	1	2	1
2	1	1	2
1	3	4	1
2	2	1	3

Dense Conv

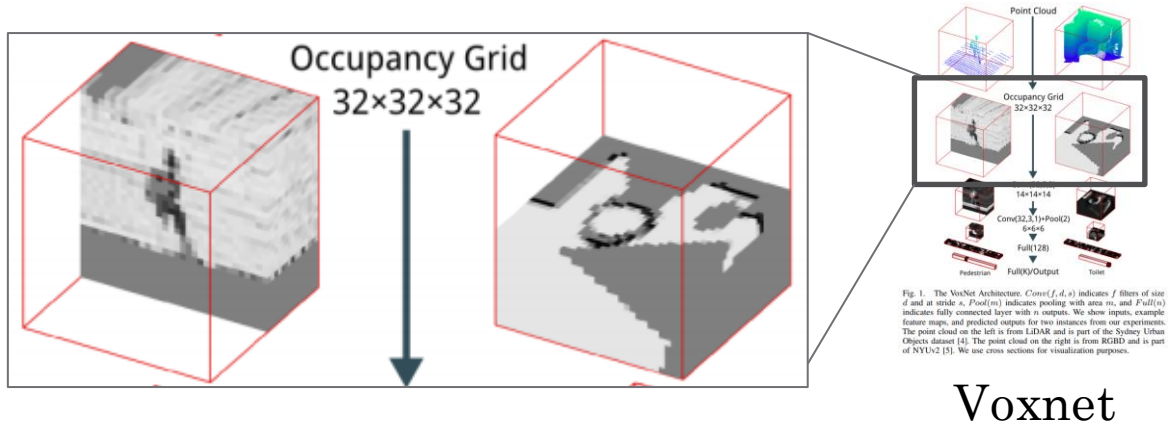
1			
	2		1
3			
	4	5	
			1

Sparse Conv

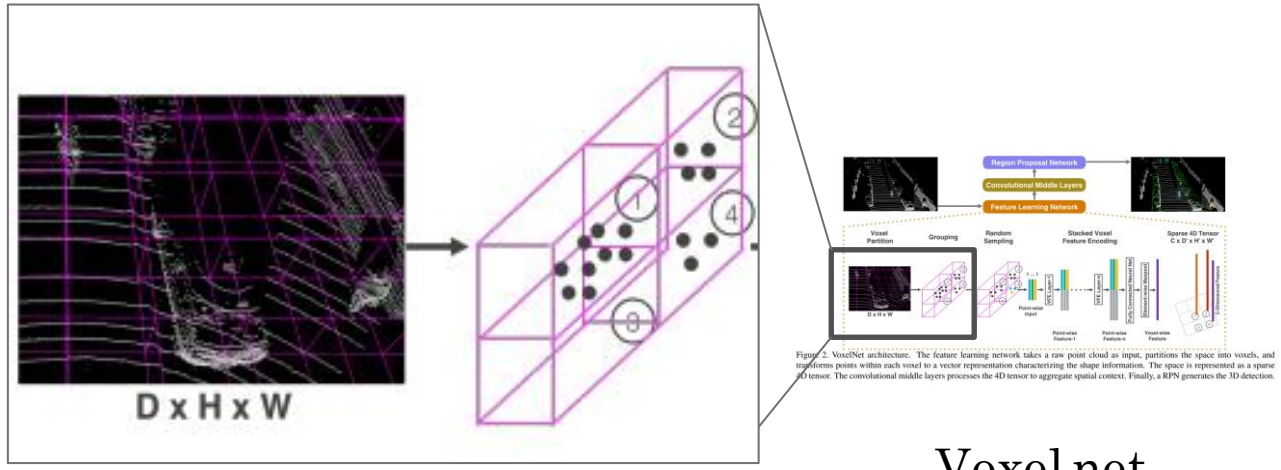
G. Riegler, A. O. Ulusoy and A. Geiger, "OctNet: Learning Deep 3D Representations at High Resolutions," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 6620-6629, doi: 10.1109/CVPR.2017.701.

2. Volumetric Representation

- Pros
  - Computationally efficient
  - Can consider local and global geometry of object in 3D space
- Cons
  - Loss of information due to quantization
  - Cannot train voxel resolution



Voxnet

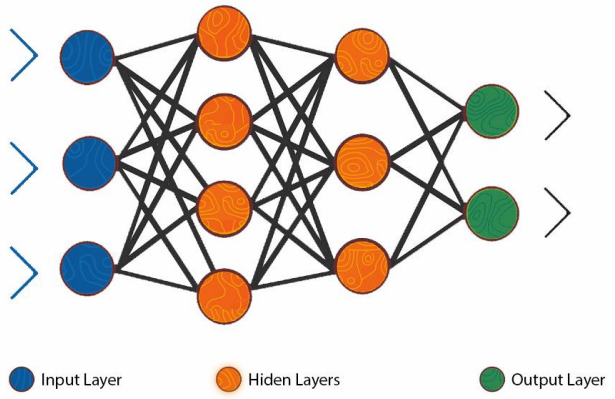


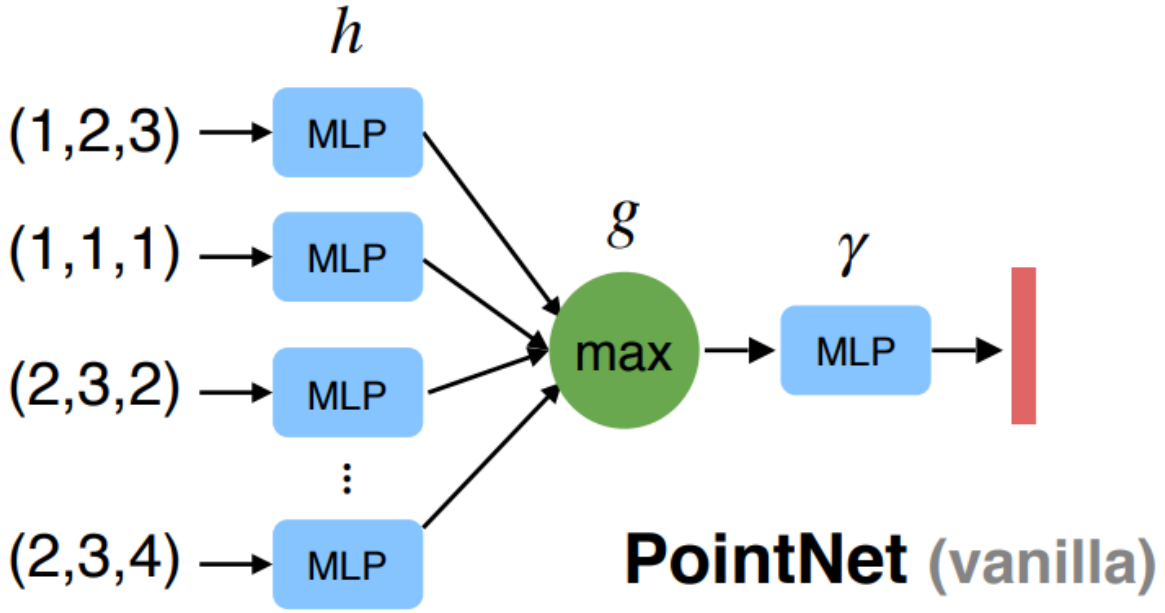
Voxel net

D. Maturana and S. Scherer, "VoxNet: A 3D Convolutional Neural Network for real-time object recognition," 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 922-928, doi: 10.1109/IROS.2015.7353481.Y.
Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4490-4499, doi: 10.1109/CVPR.2018.00472.

3a. Point Based (MLPs)

- Basic idea: (Multi-Layer Perceptrons) directly operate on point clouds
- Served as basic building blocks for other types of networks to learn pointwise features



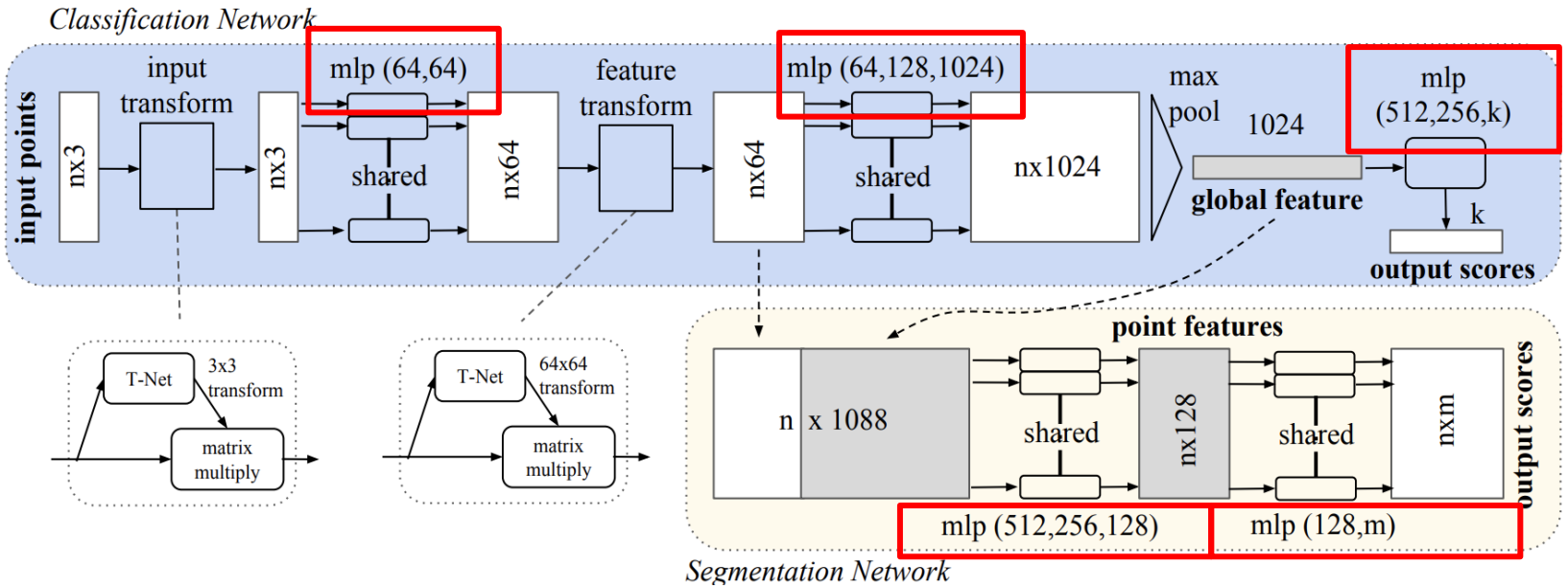


PointNet use mlp and max pooling to achieve permutation invariance

Image source: https://cdn.analyticsvidhya.com/wp-content/uploads/2020/02/ANN-Graph.gif

Image source: https://web.stanford.edu/~rqi/pointnet/docs/cvpr17_pointnet_slides.pdf

3a. Point Based (MLPs)



- Pros:
  - Feature learning directly on point cloud
  - Avoid information loss – learn per point features
  - Can consider every single point
  - Efficiently deal with unstructured data without having a grid
- Cons:
  - MLP operation does not consider spatial relationsip
  - Very hard to capture local shape in simple and efficient way
  - Computationally more expensive than volumetric representation

3b. Point Based (Convolution)

- Basic idea: Convolution 1) using neighboring points 2) continuous convolution 3) discrete convolution depends on the design of the kernel

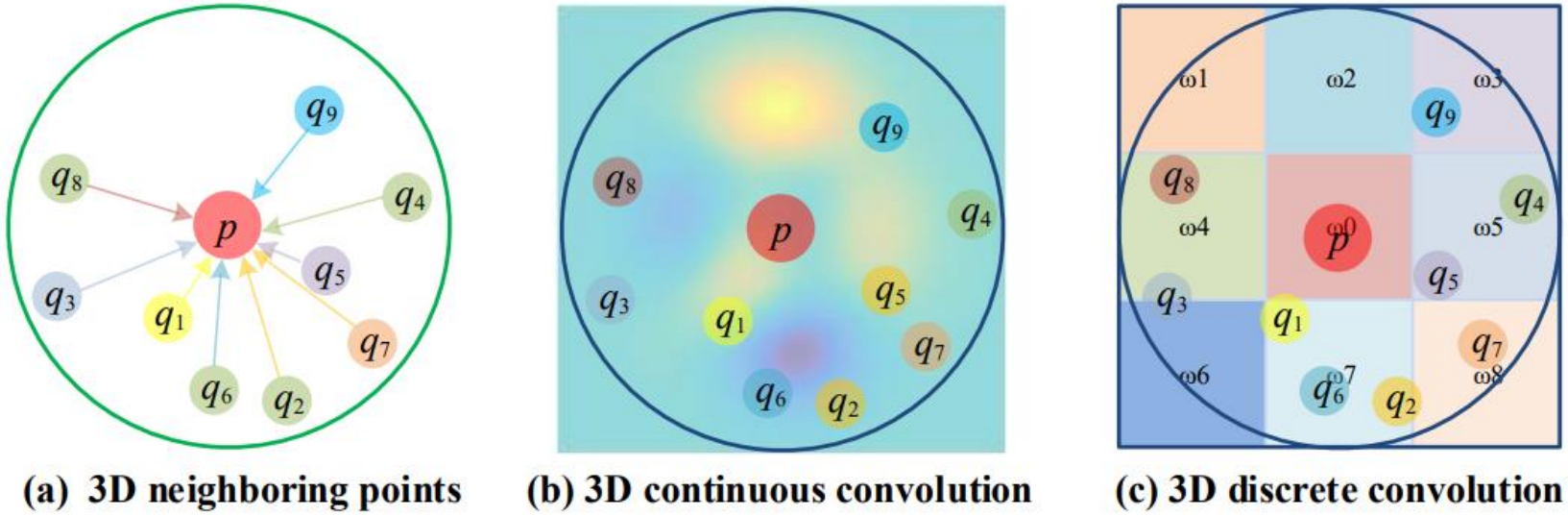
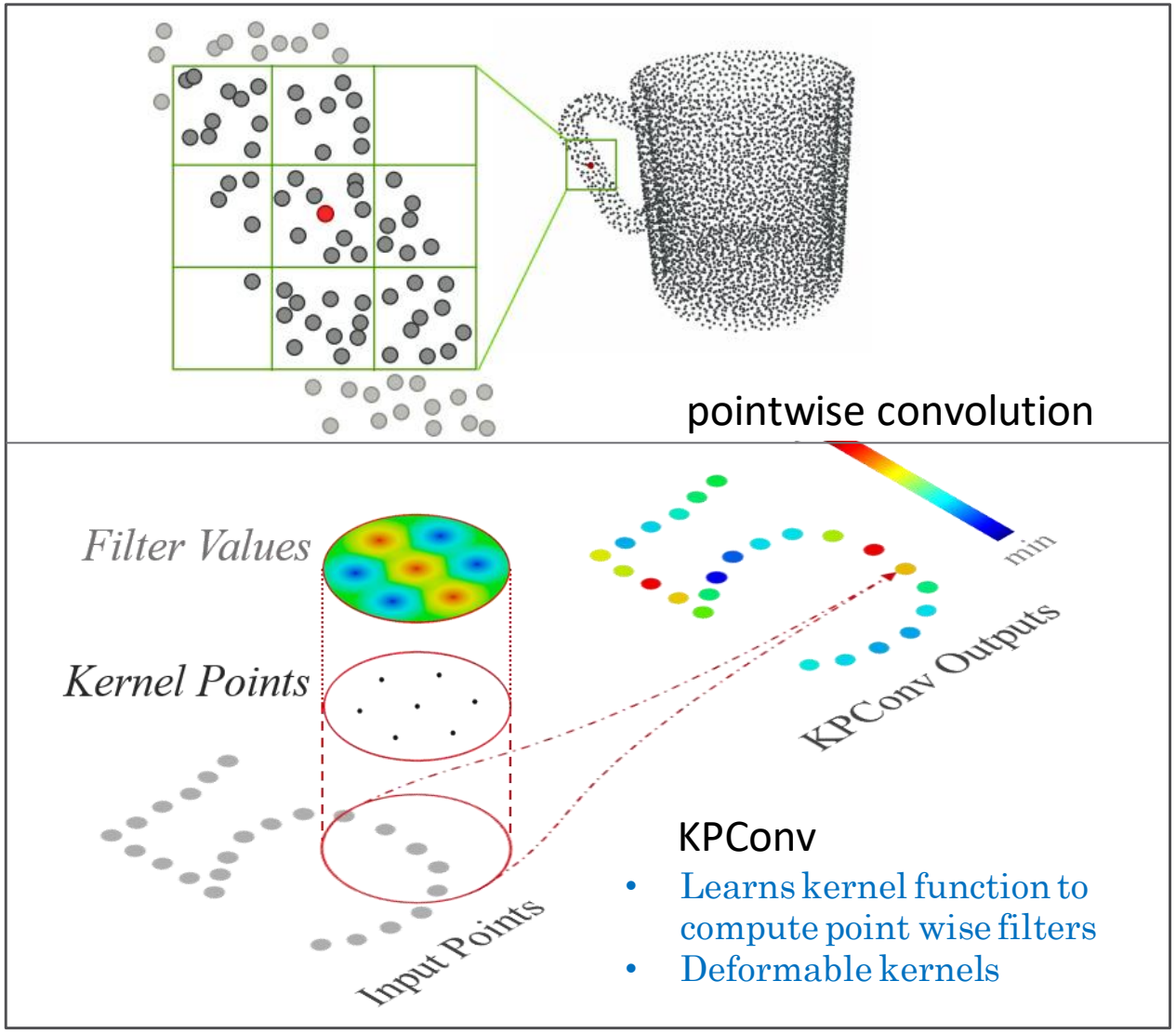


Fig. 4: An illustration of a continuous and discrete convolution for local neighbors of a point. (a) represents a local neighborhood q_i centered at point p; (b) and (c) represent 3D continuous and discrete convolution, respectively.

Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun, 2020 "Deep Learning for 3D Point Clouds: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2020.3005434.

3b. Point Based (Convolution)

- Examples: pointwise conv, KPConv
- Pros
  - Based on sampling of points so it is faster than mlps
  - Consider local geometry
- Cons
  - Sensitive to kernel design
  - Sensitive to design of sampling

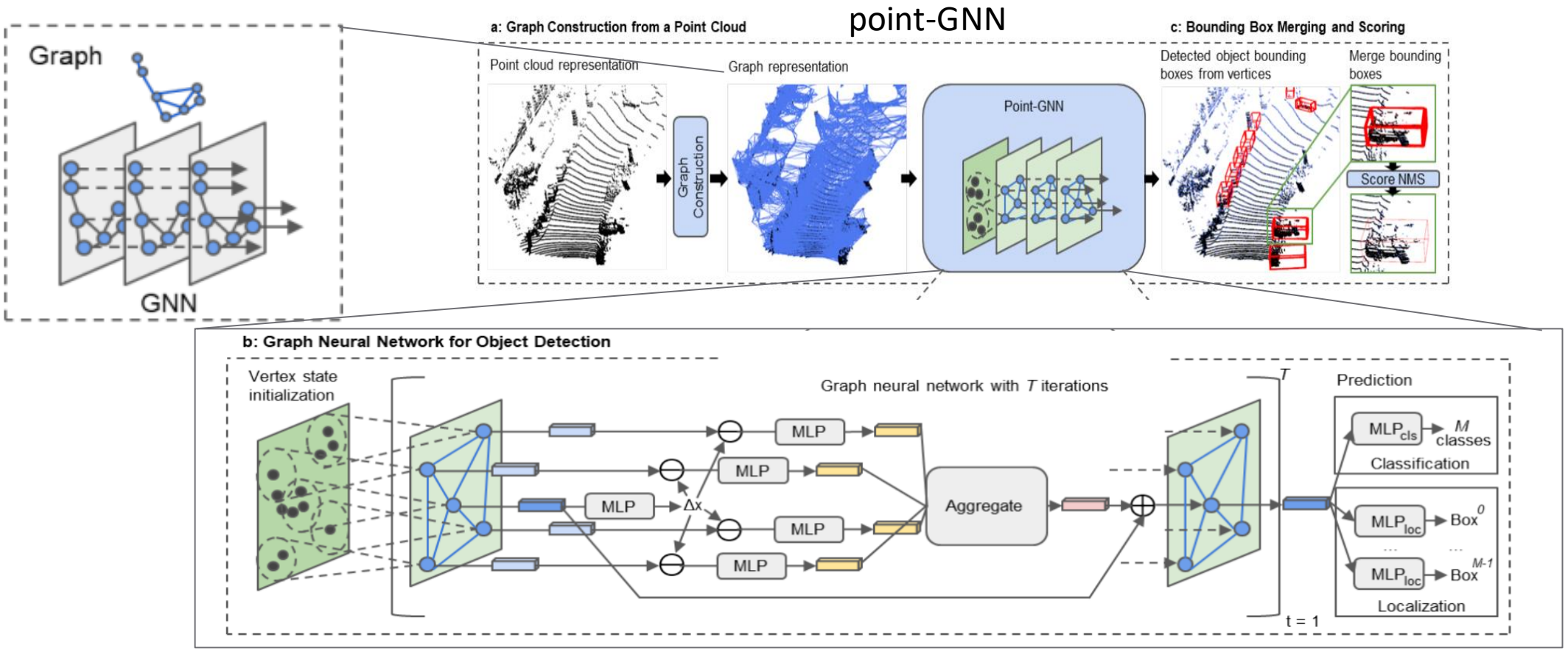


B. Hua, M. Tran and S. Yeung, "Pointwise Convolutional Neural Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 984-993, doi: 10.1109/CVPR.2018.00109.

H. Thomas, et al., "KPConv: Flexible and Deformable Convolution for Point Clouds," in 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea (South), 2019 pp. 6410-6419. doi: 10.1109/ICCV.2019.00651

4. Graph Based

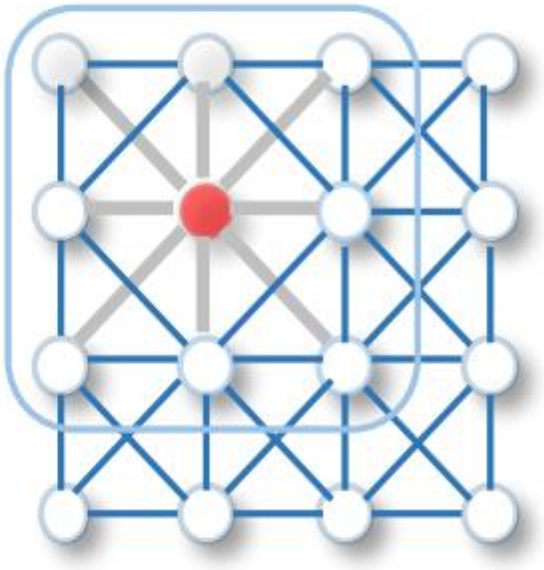
- Basic idea: Consider each point as a vertex of a graph, connected by edges



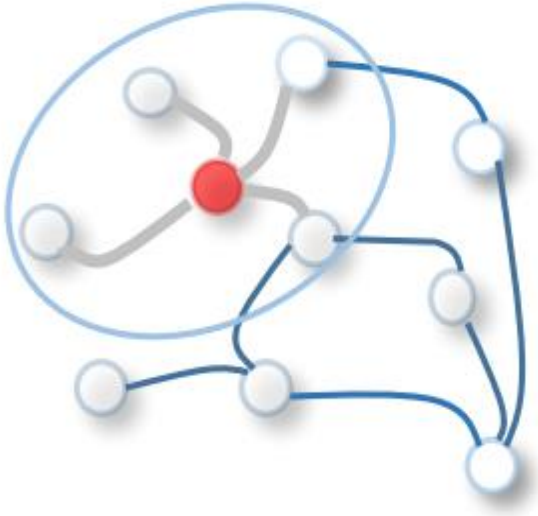
Weijing Shi, Raj Rajkumar; Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 1711-1719

4. Graph Based

- Basic idea: Consider each point as a vertex of a graph, connected by edges



(a) 2D Convolution

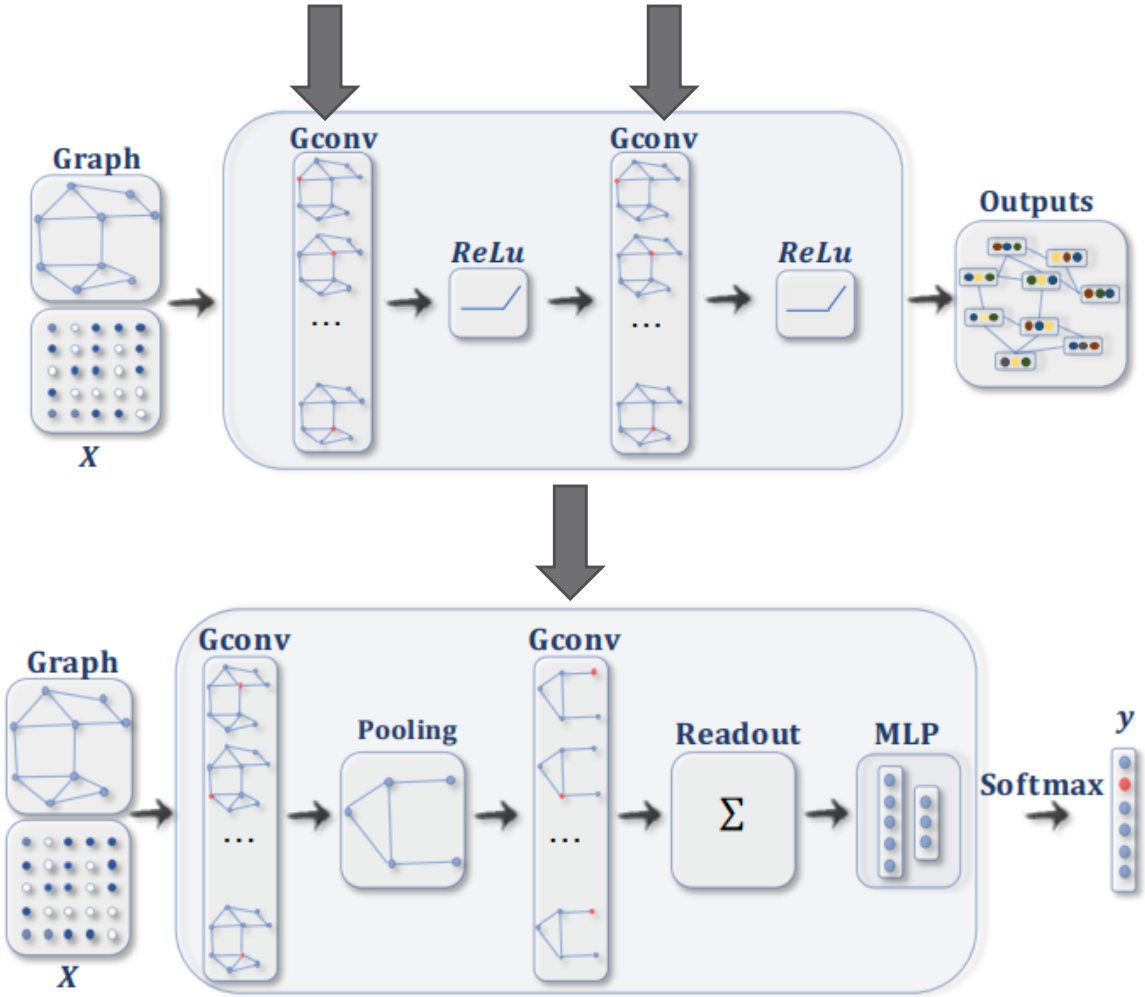


(b) Graph Convolution

Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4-24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.

4. Graph Based

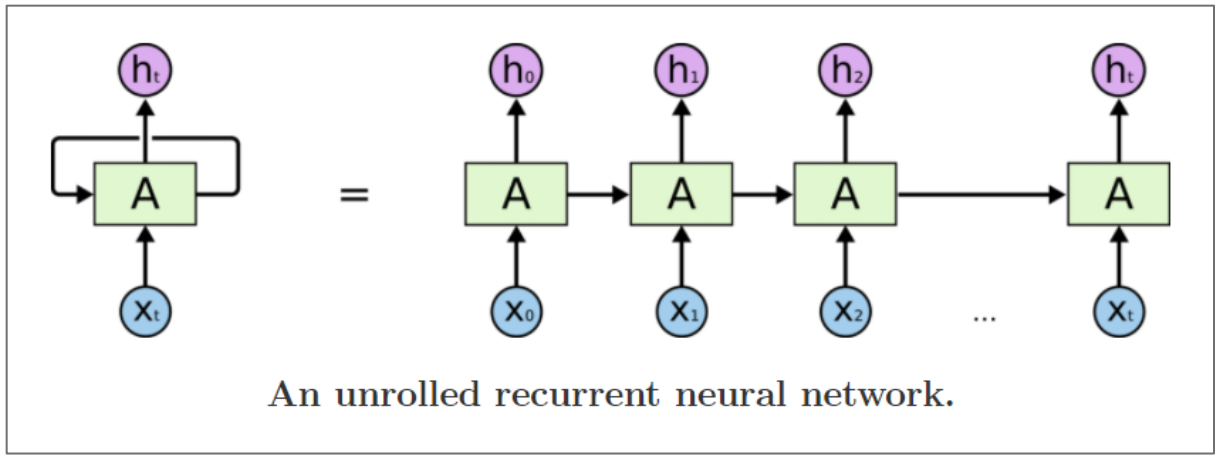
- Pros:
  - Consider topology of the points directly and high potential in solving ordering problem
  - Better understanding of shapes and geometric structures of point cloud
- Cons:
  - Graph construction normally based on a subsample of points, so depend on the method of partition
  - Feature aggregation over edges ignores local shape, focuses on surface representation



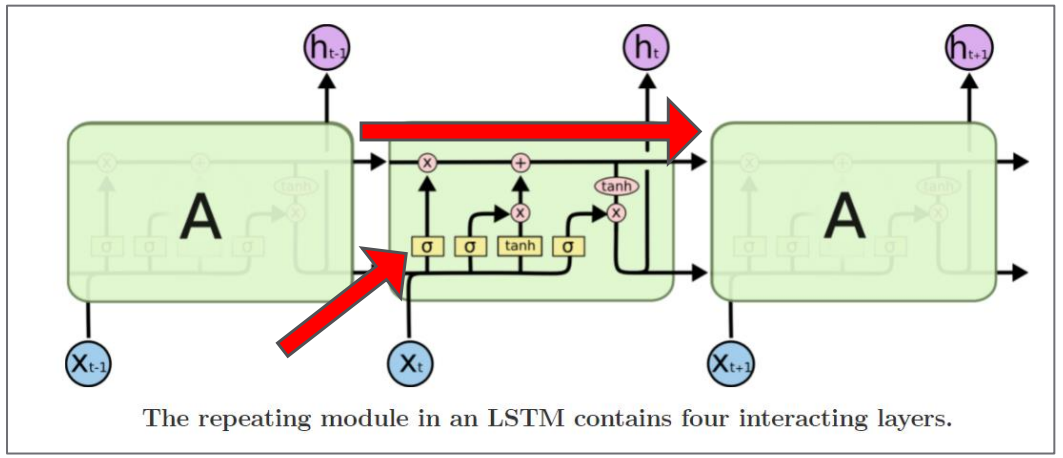
Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," in IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4-24, Jan. 2021, doi: 10.1109/TNNLS.2020.2978386.

5. RNN Based

- Basic idea: Getting popular in semantic segmentation tasks, all RNNs have feedback loops in the recurrent layer for maintaining the information 'memory' over time, and then predict the next step.



An unrolled recurrent neural network.



The repeating module in an LSTM contains four interacting layers.

LSTM (Long Short Term Memory)

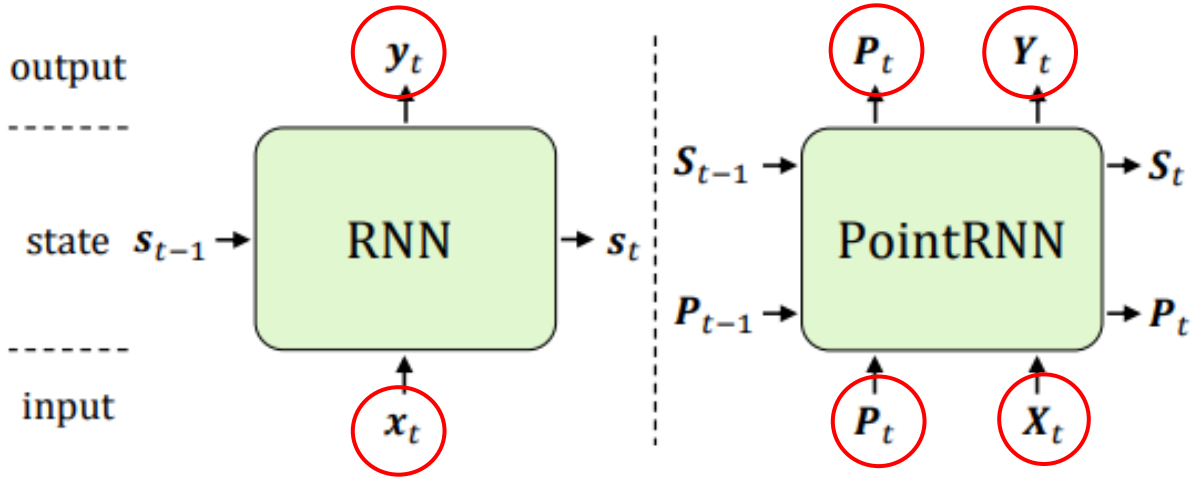
Working memory

Long term memory +
Working memory

Image source: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

5. RNN Based

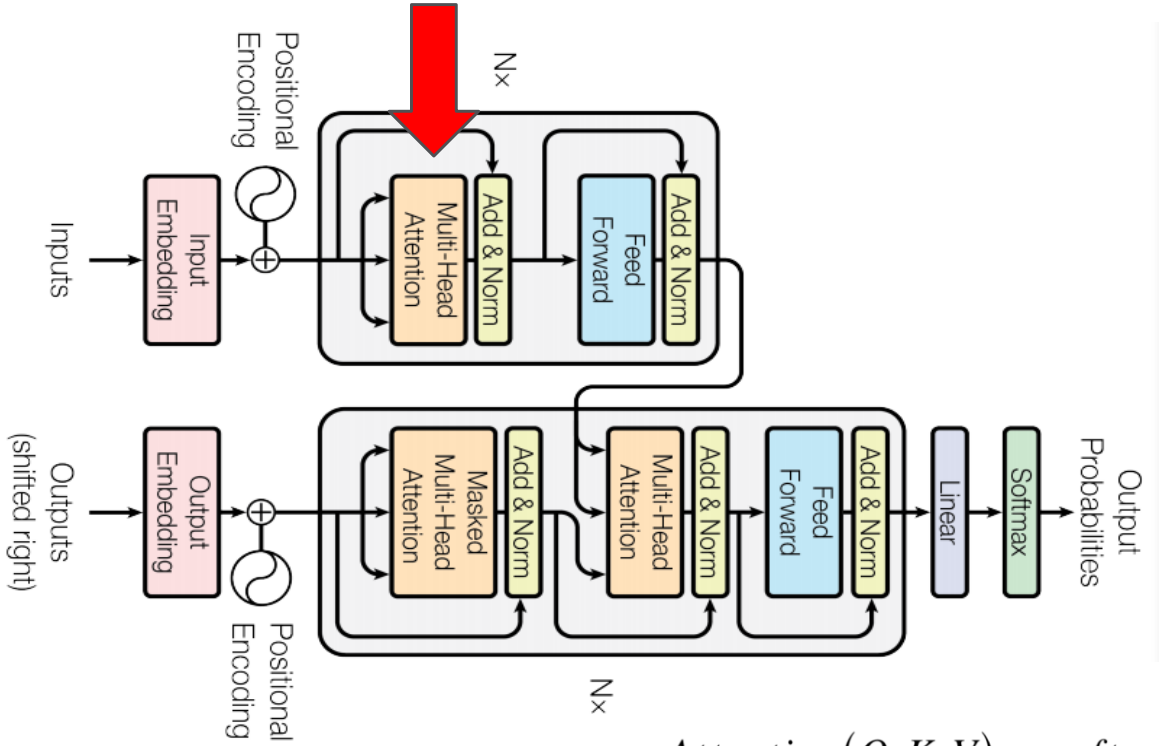
- Pros:
  - Get does not depend on convolution
  - * Lead to attention-based methods
- Cons
  - Slow, even slower for LSTM
  - Break into small pieces and loss geometric features



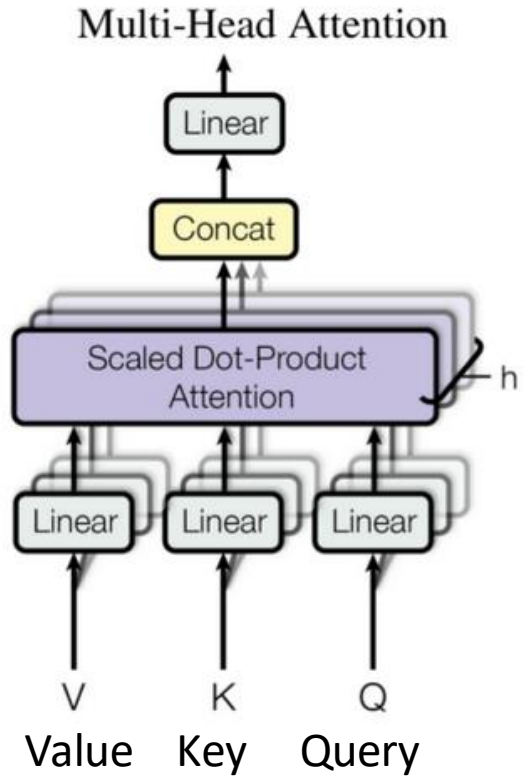
Fan, H., & Yang, Y. (2019). PointRNN: Point Recurrent Neural Network for Moving Point Cloud Processing. arXiv, abs/1910.08287. https://arxiv.org/abs/1910.08287

Bonus: Attention and Transformer

- The input data for RNN and LSTM need to be passed sequentially or serially
- Need previous state to operate on current state, not efficient for GPU for computation (parallel)



$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$



Vaswani, Ashish & Shazeer, Noam & Parmar, Niki & Uszkoreit, Jakob & Jones, Llion & Gomez, Aidan & Kaiser, Lukasz & Polosukhin, Illia, "Attention is all you need", 2017.

Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. arXiv preprint arXiv:2101.01169, 2021

Bonus: Attention and Transformer

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

The FBI is chasing a criminal on the run .

A woman is throwing a frisbee in a park.

A dog is standing on a hardwood floor.

A stop sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of people sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.

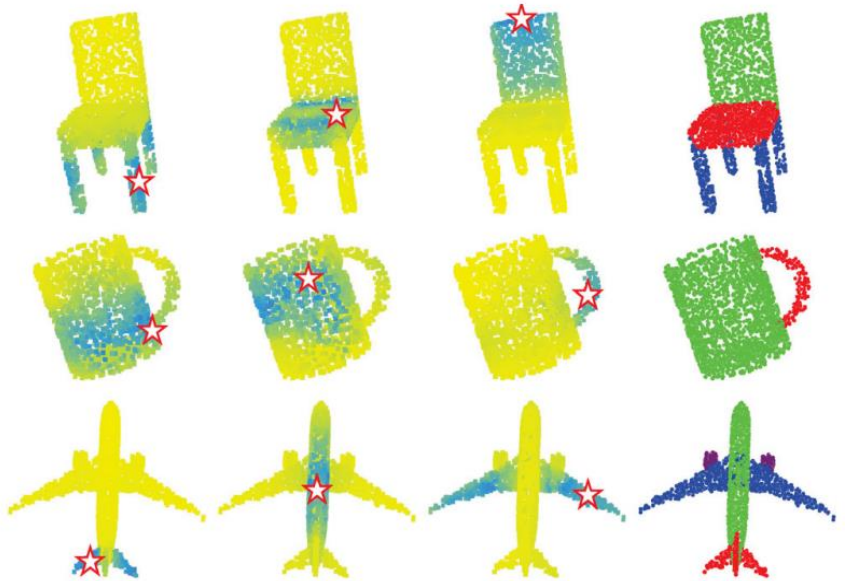


Fig. 1 Attention map and part segmentation generated by PCT. First three columns: point-wise attention map for different query points (indicated by ☆), yellow to blue indicating increasing attention weight. Last column: part segmentation results.

Guo, MH., Cai, JX., Liu, ZN. *et al.* PCT: Point cloud transformer. *Comp. Visual Media* **7,** 187–199 (2021). https://doi.org/10.1007/s41095-021-0229-5

Image source: https://shairozsohail.medium.com/a-survey-of-visual-attention-mechanisms-in-deep-learning-1043eb25f343 https://www.analyticsvidhya.com/blog/?utm_source=home_blog_navbar

3D Classification Networks

Point based

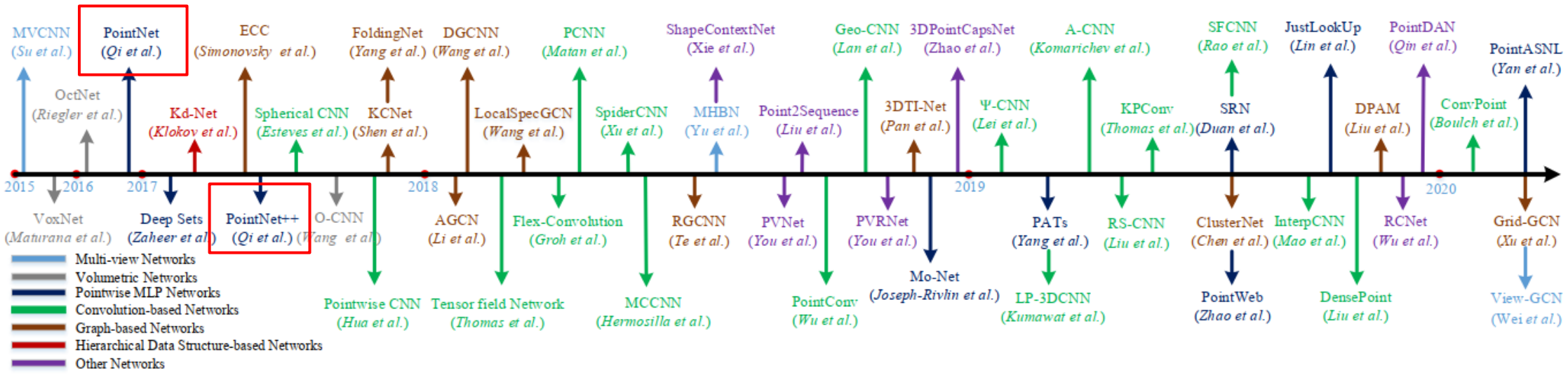


Fig. 2: Chronological overview of the most relevant deep learning-based 3D shape classification methods.

Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun, 2020 "Deep Learning for 3D Point Clouds: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2020.3005434.

3D Detection Networks

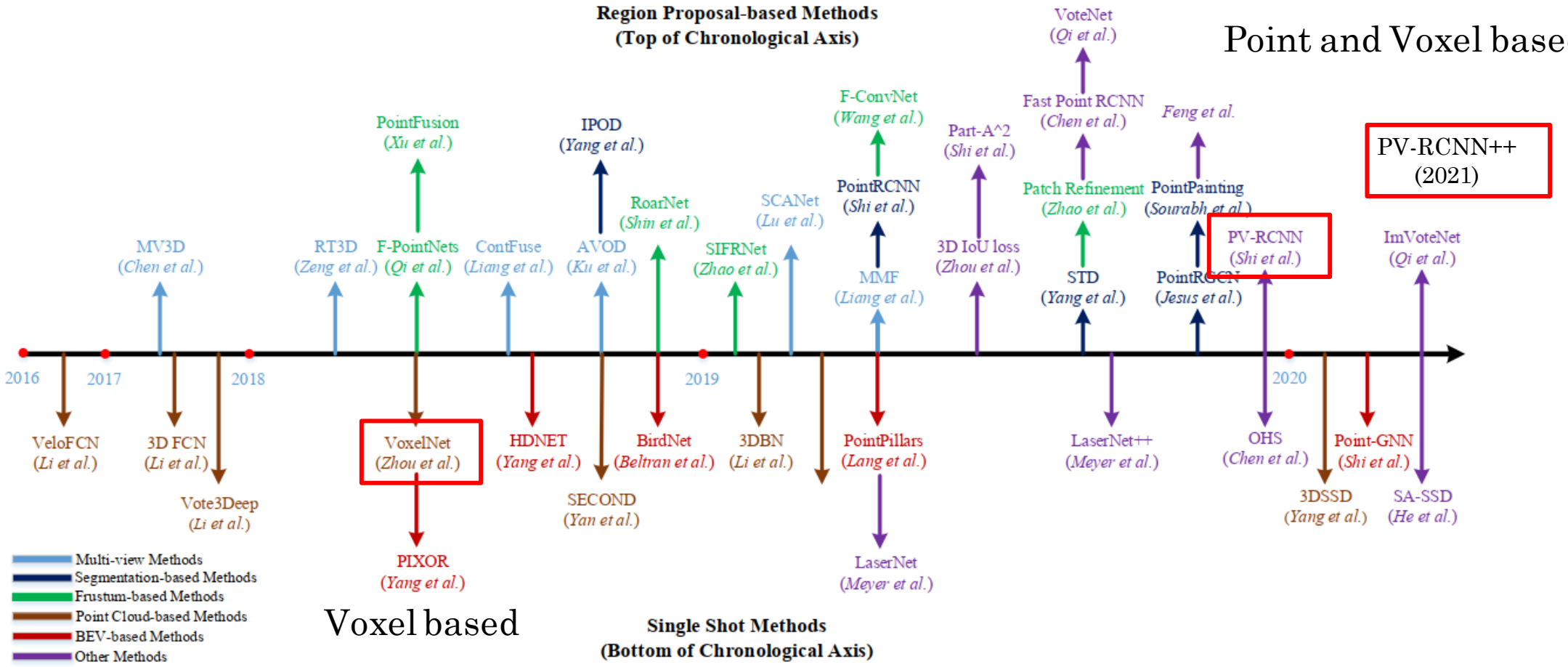


Fig. 7: Chronological overview of the most relevant deep learning-based 3D object detection methods.

Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu and M. Bennamoun, 2020 "Deep Learning for 3D Point Clouds: A Survey," in IEEE Transactions on Pattern Analysis and Machine Intelligence,doi: 10.1109/TPAMI.2020.3005434.

PointNet

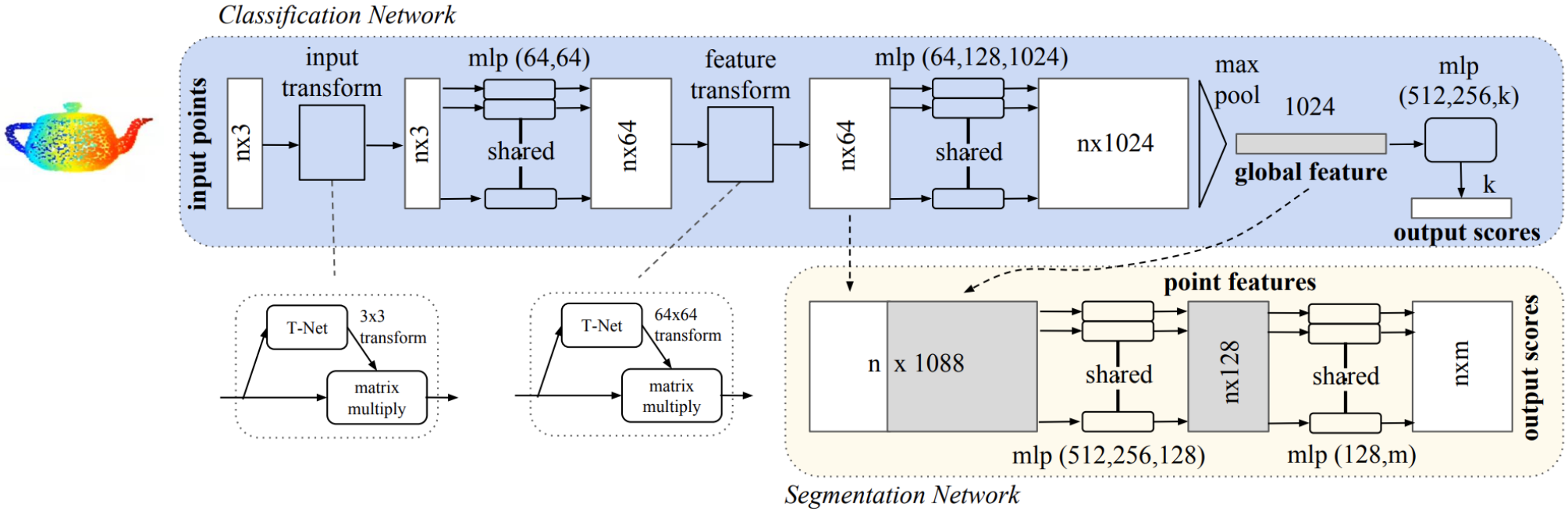
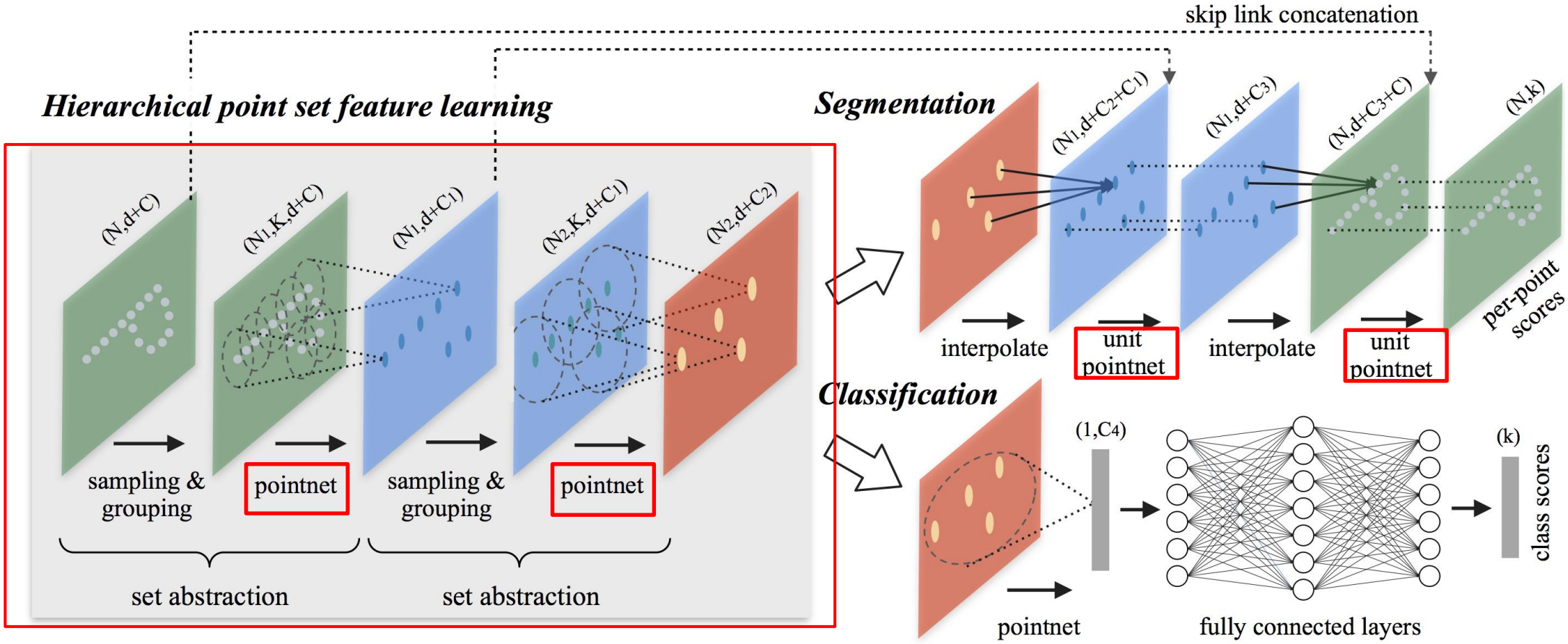


Figure 2. **PointNet Architecture.** The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

Charles R.Q., Su H., Kaichun M., Guibas L.J. Pointnet: Deep Learning on Point Sets for 3d Classification and Segmentation; Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); Honolulu, HI, USA. 21–26 July 2017; pp. 77–85. https://arxiv.org/abs/1612.00593

PointNet ++



Qi R., Yi L., Su H., Guibas L.J. Pointnet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space; Proceedings of the Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems; Long Beach, CA, USA. 4–9 December 2017.

VoxelNet

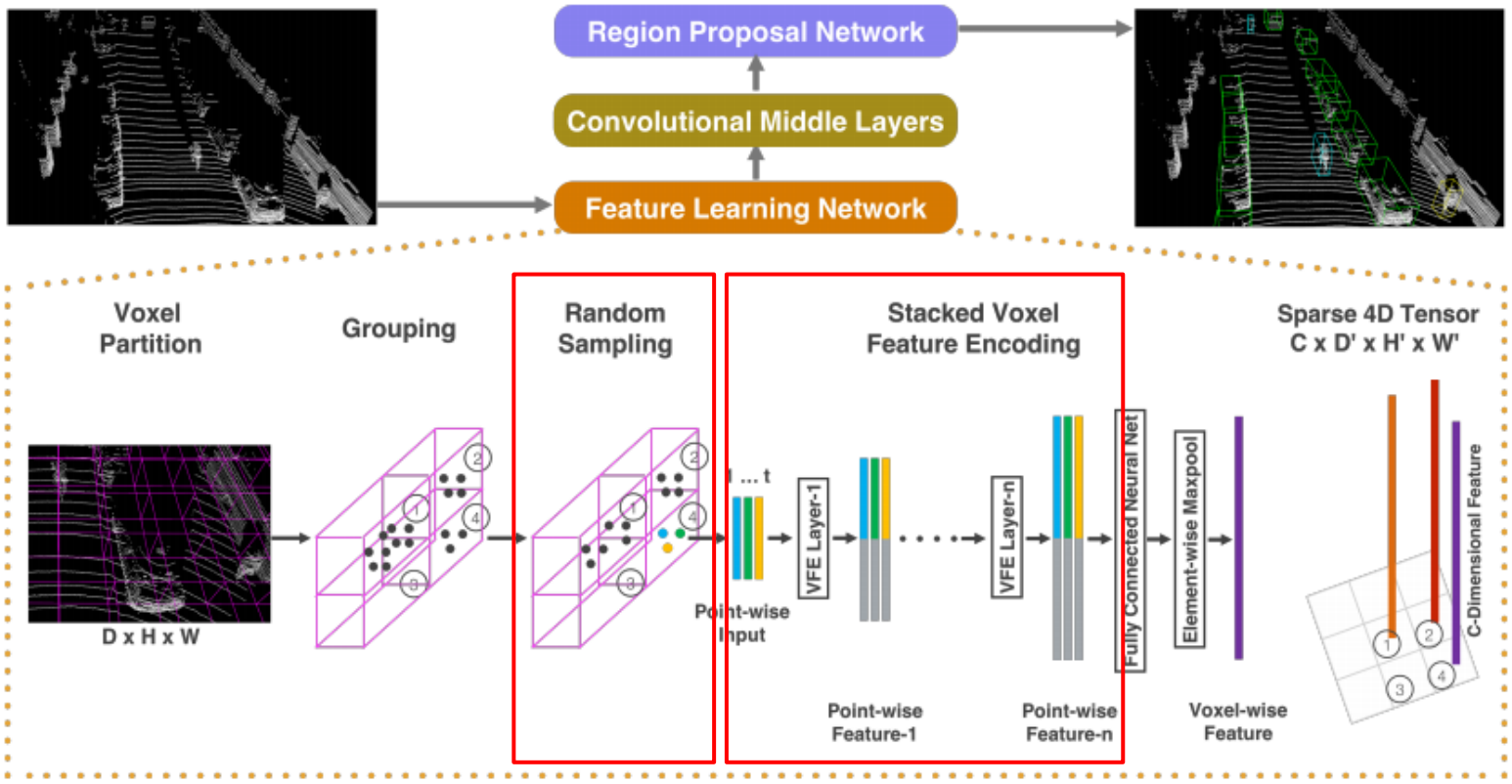


Figure 2. VoxelNet architecture. The feature learning network takes a raw point cloud as input, partitions the space into voxels, and transforms points within each voxel to a vector representation characterizing the shape information. The space is represented as a sparse 4D tensor. The convolutional middle layers processes the 4D tensor to aggregate spatial context. Finally, a RPN generates the 3D detection.

Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 4490-4499, doi: 10.1109/CVPR.2018.00472.https://arxiv.org/pdf/1711.06396.pdf

VoxelNet

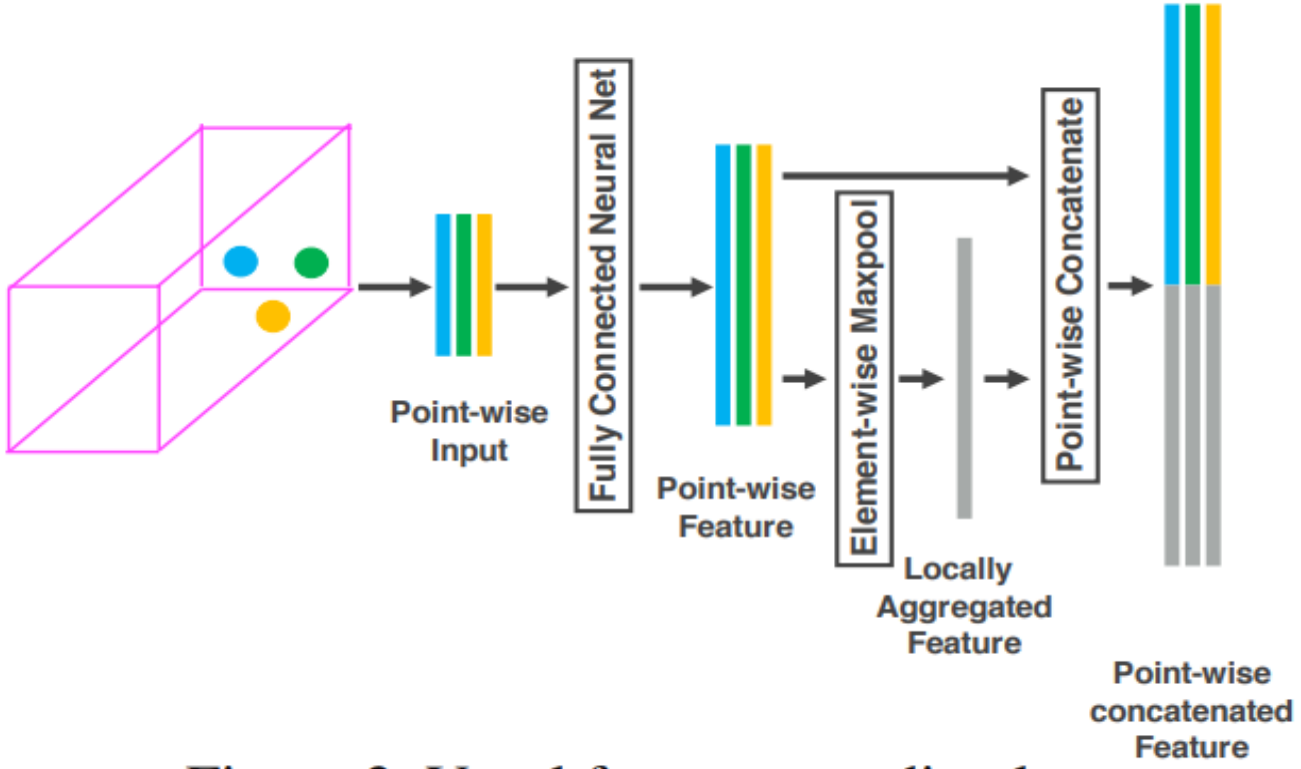


Figure 3. Voxel feature encoding layer.

PV-RCNN

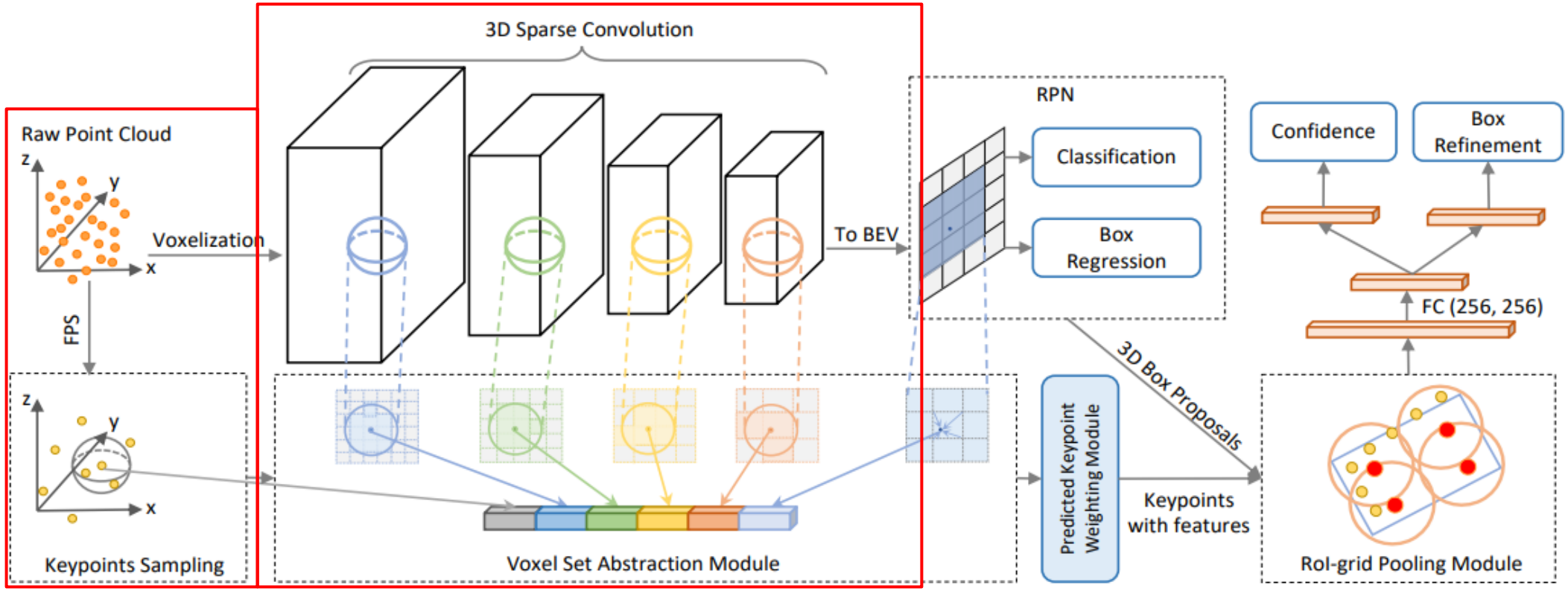


Figure 2. The overall architecture of our proposed PV-RCNN. The raw point clouds are first voxelized to feed into the 3D sparse convolution based encoder to learn multi-scale semantic features and generate 3D object proposals. Then the learned voxel-wise feature volumes at multiple neural layers are summarized into a small set of key points via the novel voxel set abstraction module. Finally the keypoint features are aggregated to the RoI-grid points to learn proposal specific features for fine-grained proposal refinement and confidence prediction.

S. Shi et al., "PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 10526-10535, doi: 10.1109/CVPR42600.2020.01054.https://arxiv.org/pdf/1912.13192.pdf

PV-RCNN ++

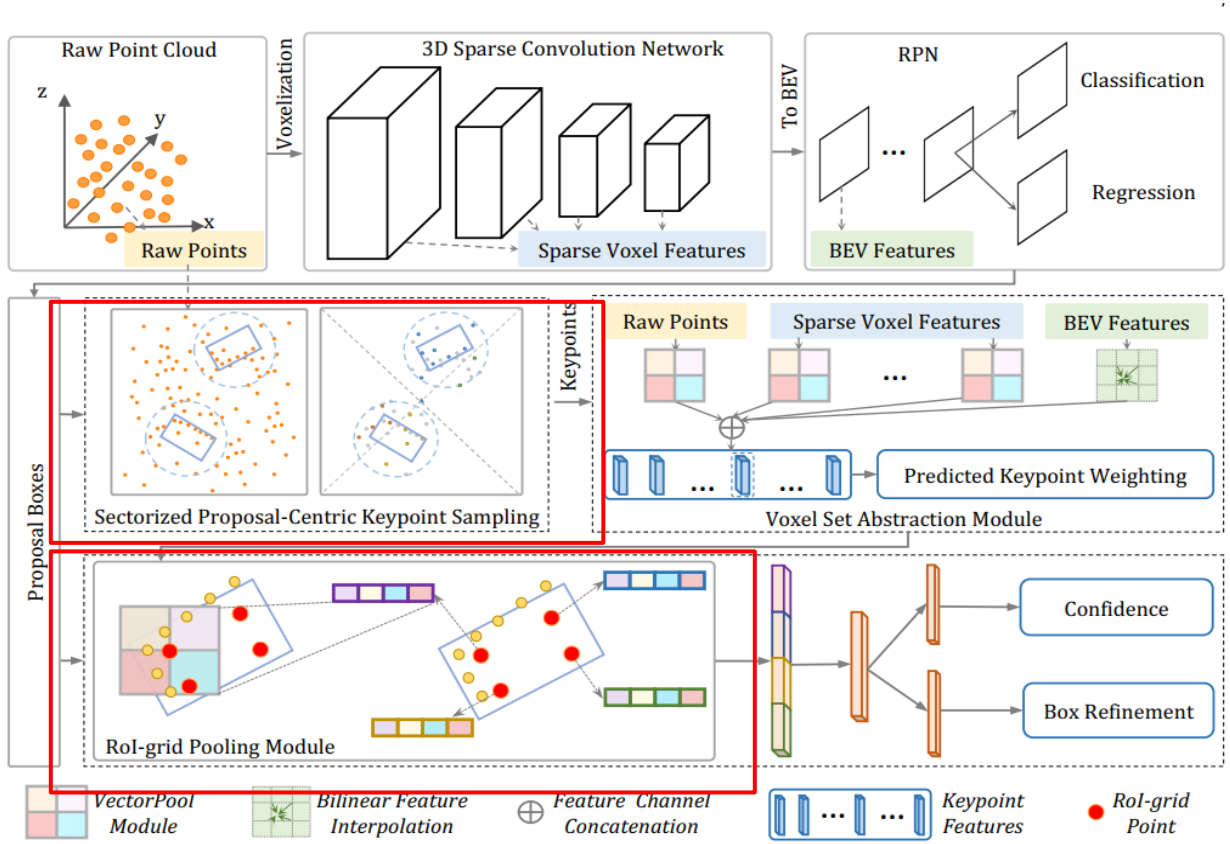
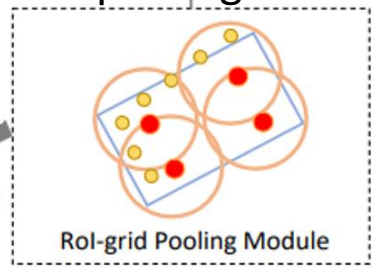


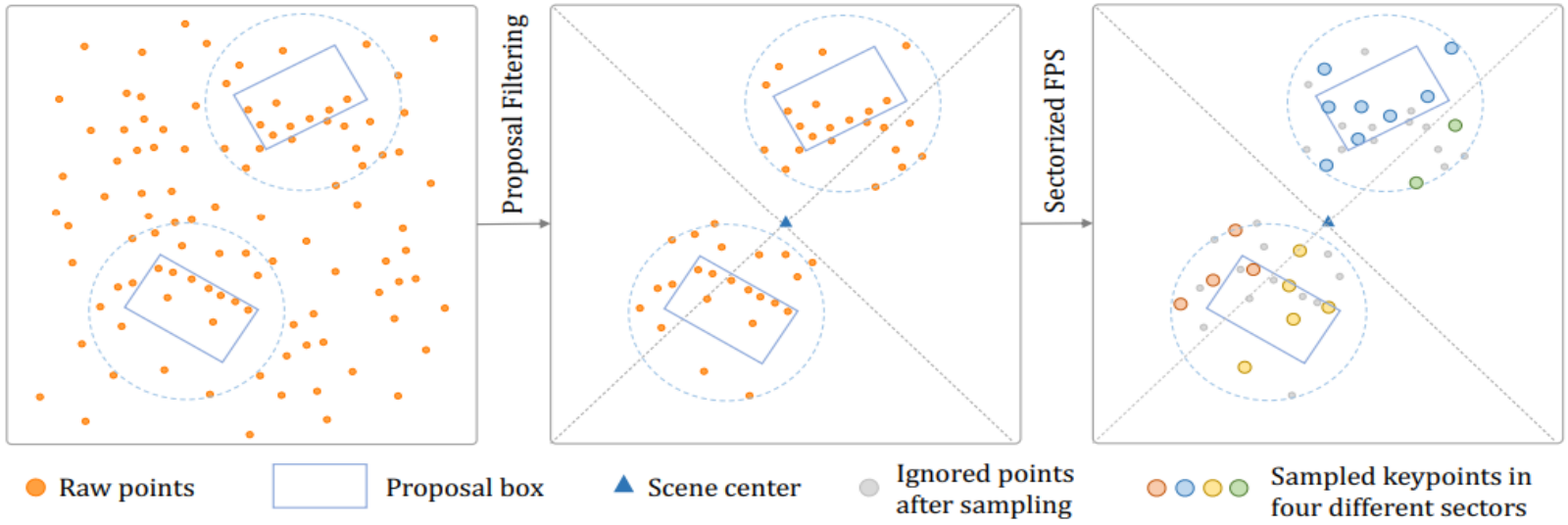
Fig. 3. The overall architecture of our proposed PV-RCNN-v2. We propose the Sectorized-Proposal-Centric keypoint sampling module to concentrate keypoints to the neighborhoods of 3D proposals while also accelerating the process with sectorized FPS. Moreover, our proposed VectorPool module is utilized in both the voxel set abstraction module and the RoI-grid pooling module to improve the local feature aggregation and save memory/computation resources

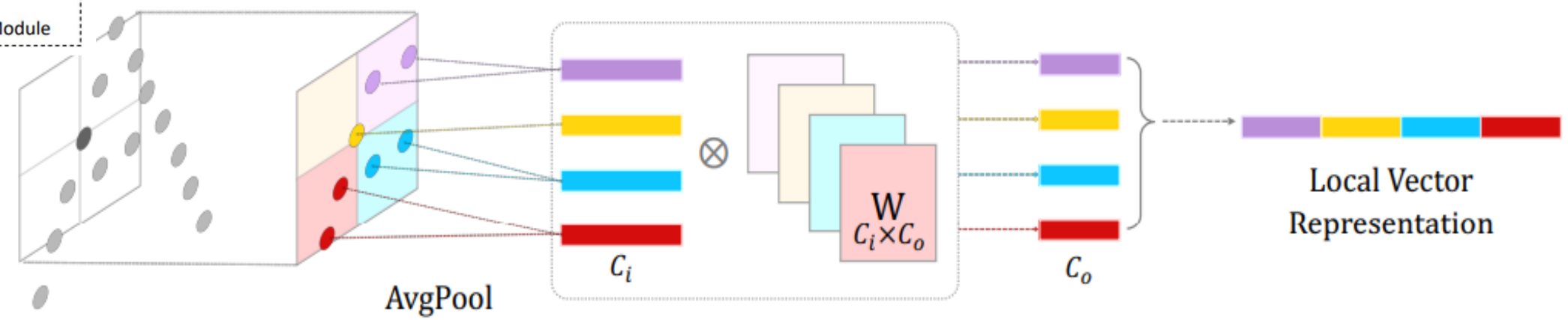
Shaoshuai Shi, Li Jiang, Jiajun Deng, Zhe Wang, Chaoxu Guo, Jianping Shi, Xiaogang Wang, Hongsheng Li. PV-RCNN++: Point-Voxel Feature Set Abstraction With Local Vector Representation for 3D Object Detection. 2021. https://arxiv.org/pdf/2102.00463.pdf

PV-RCNN ++

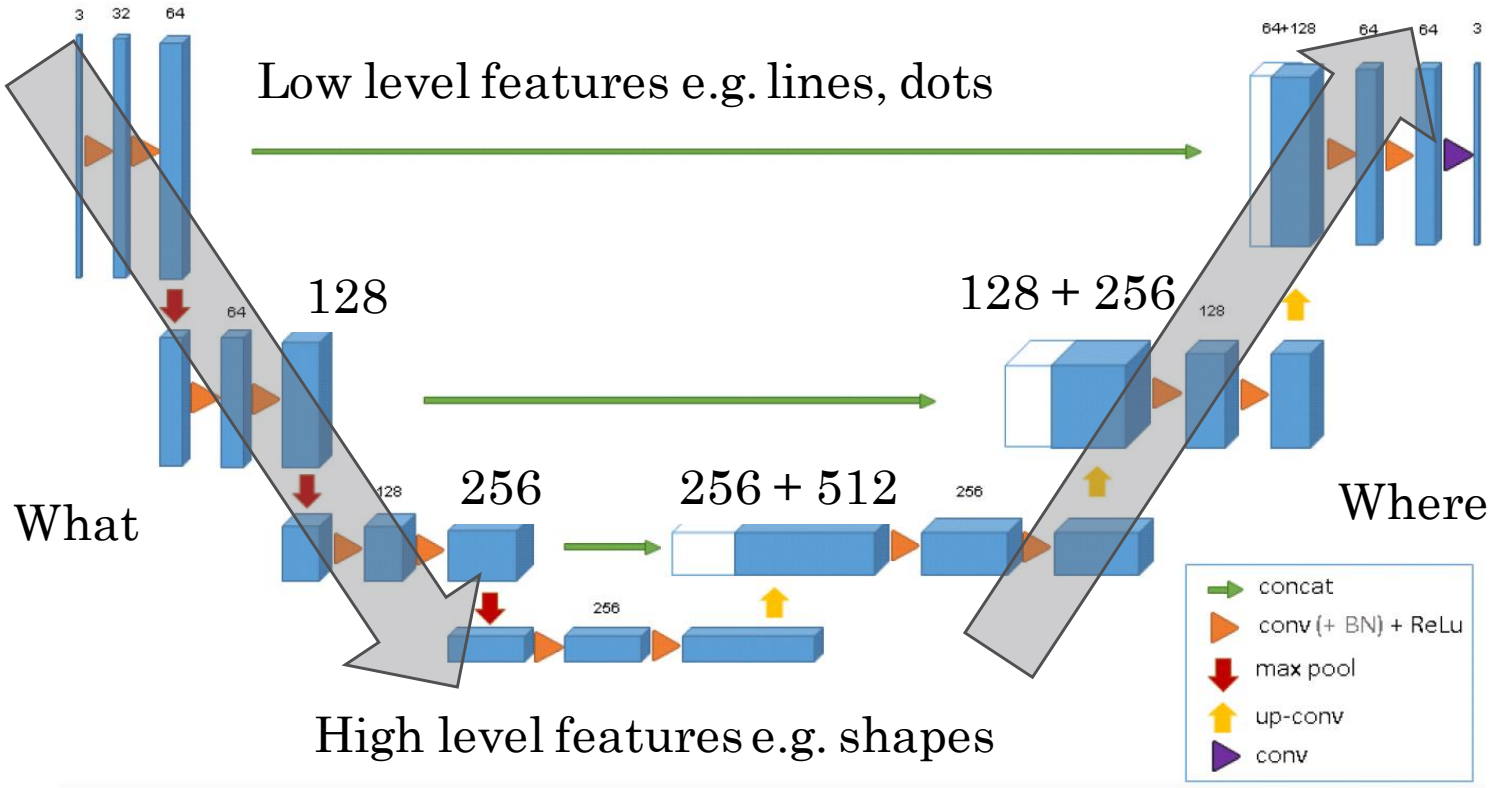
PV-RCNN pooling

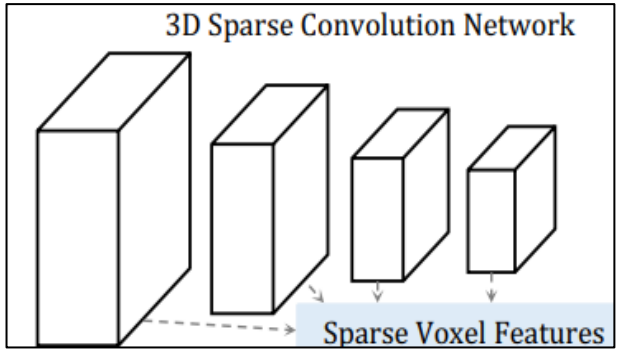






3D U-Net





Recap: Convolution Network – dimension reduction

- Why pool?
- Down-sample only? What's the problem?

Çiçek Ö., Abdulkadir A., Lienkamp S.S., Brox T., Ronneberger O. (2016) 3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation. In: Ourselin S., Joskowicz L., Sabuncu M., Unal G., Wells W. (eds) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016. MICCAI 2016. Lecture Notes in Computer Science, vol 9901. Springer, Cham. https://doi.org/10.1007/978-3-319-46723-8_49

Benchmarking Datasets

Datasets for 3D Shape Classification							
Name and Reference	Year	#Samples	#Classes	#Training	#Test	Type	Representation
McGill Benchmark [23]	2008	456	19	304	152	Synthetic	Mesh
Sydney Urban Objects [24]	2013	588	14	-	-	Real-World	Point Clouds
ModelNet10 [6]	2015	4899	10	3991	605	Synthetic	Mesh
ModelNet40 [6]	2015	12311	40	9843	2468	Synthetic	Mesh
ShapeNet [8]	2015	51190	55	-	-	Synthetic	Mesh
ScanNet [11]	2017	12283	17	9677	2606	Real-World	RGB-D
ScanObjectNN [7]	2019	2902	15	2321	581	Real-World	Point Clouds
Datasets for 3D Object Detection and Tracking							
Name and Reference	Year	#Scenes	#Classes	#Annotated Frames	#3D Boxes	Secne Type	Sensors
KITTI [14]	2012	22	8	15K	200K	Urban (Driving)	RGB & LiDAR
SUN RGB-D [25]	2015	47	37	5K	65K	Indoor	RGB-D
ScanNetV2 [11]	2018	1.5K	18	-	-	Indoor	RGB-D & Mesh
H3D [26]	2019	160	8	27K	1.1M	Urban (Driving)	RGB & LiDAR
Argoverse [27]	2019	113	15	44K	993K	Urban (Driving)	RGB & LiDAR
Lyft L5 [28]	2019	366	9	46K	1.3M	Urban (Driving)	RGB & LiDAR
A*3D [29]	2019	-	7	39K	230K	Urban (Driving)	RGB & LiDAR
Waymo Open [30]	2020	1K	4	200K	12M	Urban (Driving)	RGB & LiDAR
nuScenes [31]	2020	1K	23	40K	1.4M	Urban (Driving)	RGB & LiDAR
Datasets for 3D Point Cloud Segmentation							
Name and Reference	Year	#Points	#Classes[1]	#Scans	Spatial Size	RGB	Sensors
Oakland [32]	2009	1.6M	5(44)	17	-	N/A	MLS
ISPRS [33]	2012	1.2M	9	-	-	N/A	ALS
Paris-rue-Madame [34]	2014	20M	17	2	-	N/A	MLS
IQmulus [35]	2015	300M	8(22)	10	-	N/A	MLS
ScanNet [11]	2017	-	20(20)	1513	8×4×4	Yes	RGB-D
S3DIS [10]	2017	273M	13(13)	272	10×5×5	Yes	Matterport
Semantic3D [12]	2017	4000M	8(9)	15/15	250×260×80	Yes	TLS
Paris-Lille-3D [36]	2018	143M	9(50)	3	200×280× 30	N/A	MLS
SemanticKITTI [15]	2019	4549M	25(28)	23201/20351	150×100×10	N/A	MLS
Toronto-3D [37]	2020	78.3M	8(9)	4	260×350× 40	Yes	MLS
DALES [38]	2020	505M	8(9)	40	500×500×65	N/A	ALS

https://arxiv.org/pdf/1912.12033.pdf

Summary

Challenges

Processing types:

1. Multi-view
2. Volumetric representation
3. Point based
   a) Point based mlp
   b) Point based conv
4. Graph based
5. RNN based

Case studies:

- PointNet
- PointNet ++
- VoxelNet
- PV-RCNN
- PV-RCNN ++
- 3D U-Net

Extra Resources:

- Deep Learning with Python, Second Edition
  - https://www.manning.com/books/deep-learning-with-python-second-edition
- Grokking Deep Learning
  - https://www.amazon.com/Grokking-Deep-Learning-Andrew-Trask/dp/1617293709
- Dive into Deep Learning
  - http://d2l.ai/
- Using Matlab for Deep Learning
  - https://www.mathworks.com/videos/series/deep-learning-webinars.html
- Stanford University: Lecture Collection
  - https://www.youtube.com/playlist?list=PL3FW7Lu3i5JvHM8ljYj-zLfQRF3EO8sYv
- Andrew Ng's website
  - https://www.andrewng.org/video/

Acknowledgement

THE UNIVERSITY OF BRITISH COLUMBIA

Vancouver Campus

Thank you