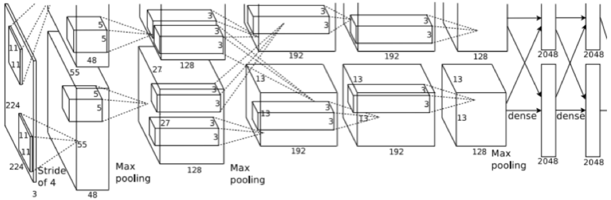
Detection and Segmentation for 2D & 3D images

Zhile Ren
Postdoc @ Georgia Tech
http://jrenzhile.com

Slides courtesy: F-F. Li, J Johnson, S. Yeung, H. Su, C. Qi

So far: Image Classification

This image is CC0 public domain



Vector: 4096

Fully-Connected: 4096 to 1000

Class Scores

Cat: 0.9

Dog: 0.05

Car: 0.01

...

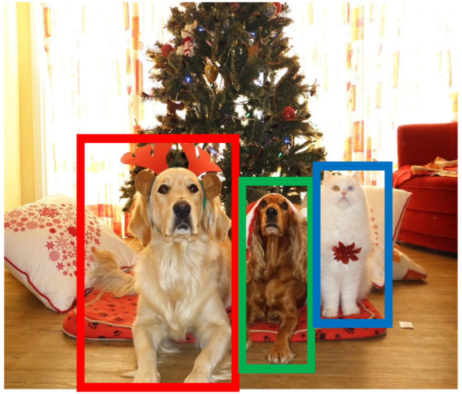
More Computer Vision Tasks

2D Semantic Segmentation

GRASS, CAT, TREE, SKY

Object categories + 2D segments

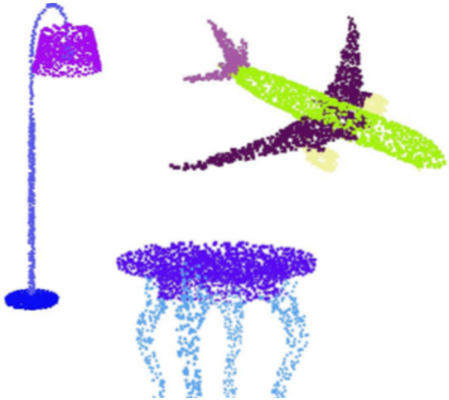
2D Object Detection



DOG, DOG, CAT

Object categories + 2D bounding boxes

3D Classificaion & Segmentation



Object categories + 3D segments

More Computer Vision Tasks

2D Semantic Segmentation

GRASS, CAT, TREE, SKY

Object categories + 2D segments

2D Object Detection

DOG, DOG, CAT

Object categories + 2D bounding boxes

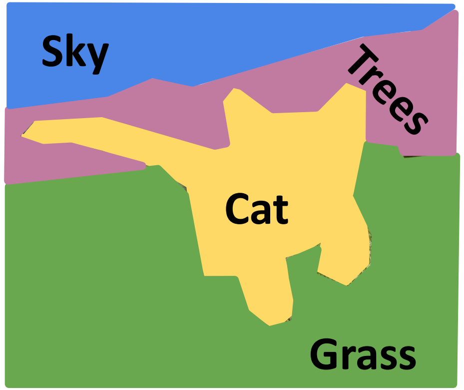
3D Classificaion & Segmentation

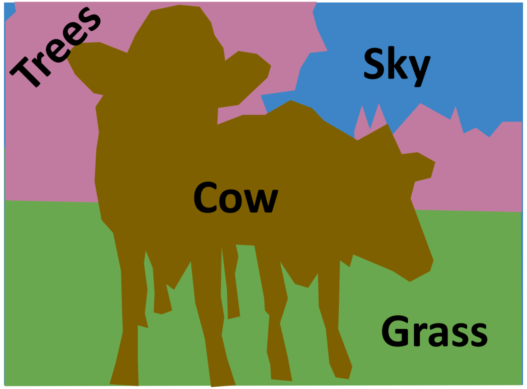
Object categories + 3D segments

Semantic Segmentation

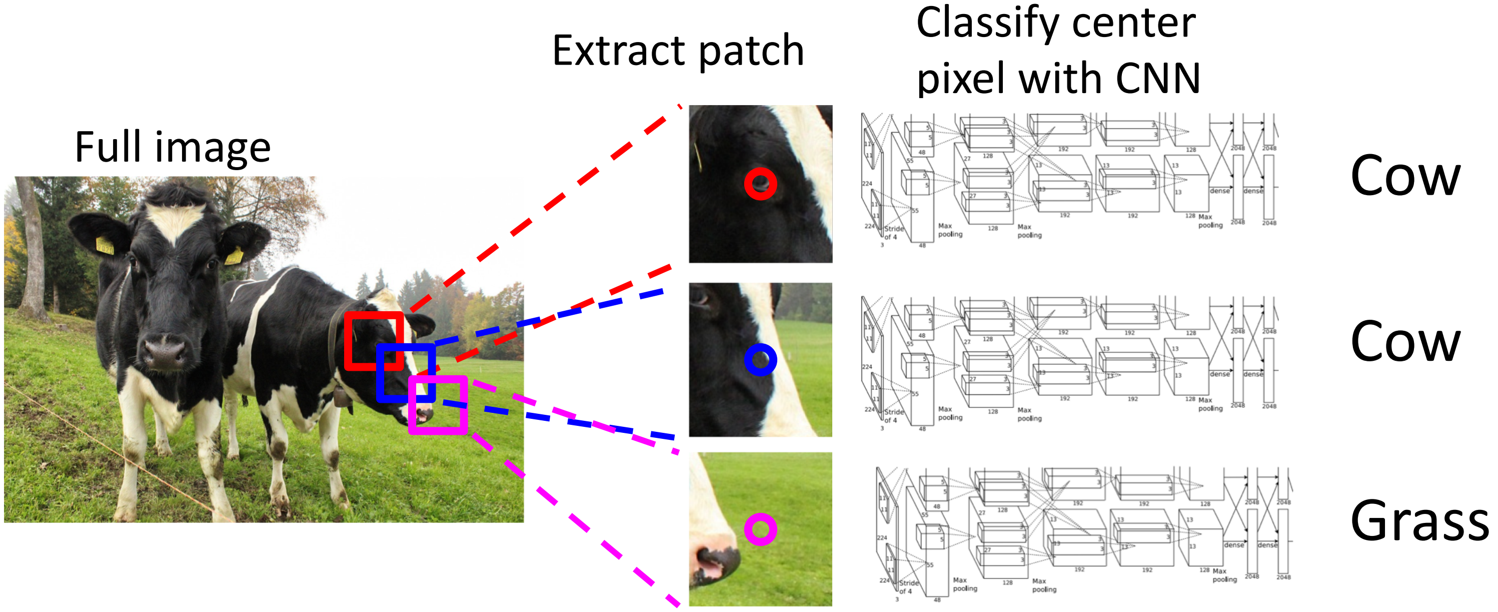
Label each pixel in the image with a category label

Don’t differentiate instances, only care about pixels





Semantic Segmentation Idea: Sliding Window



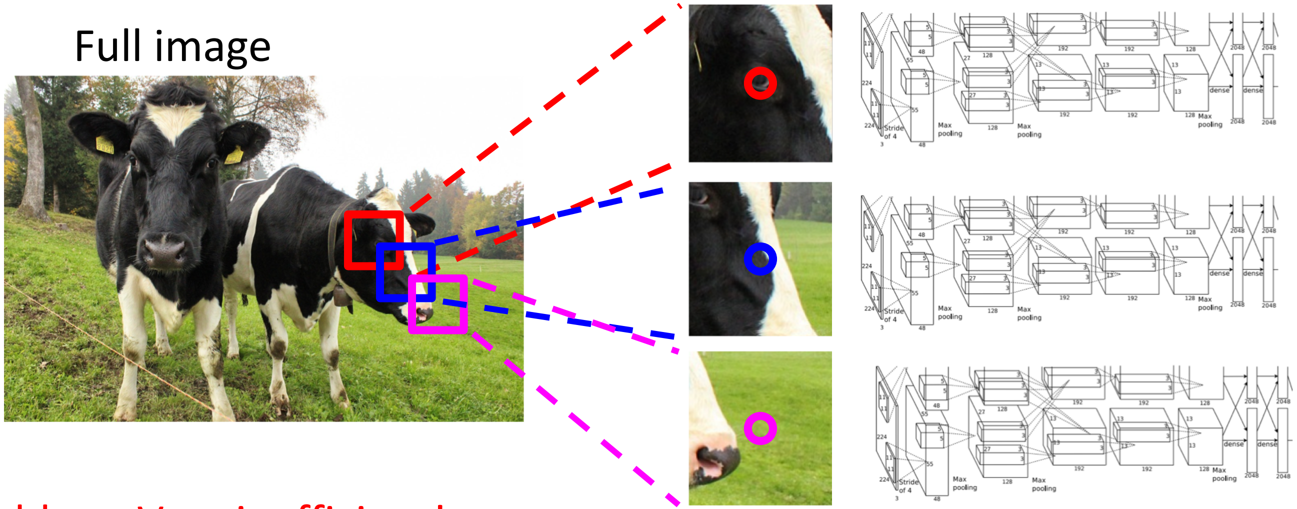
Farabet et al, “Learning Hierarchical Features for Scene Labeling,” TPAMI 2013
Pinheiro and Collobert, “Recurrent Convolutional Neural Networks for Scene Labeling”, ICML 2014

Semantic Segmentation Idea: Sliding Window

Full image

Extract patch

Classify center pixel with CNN



Cow

Cow

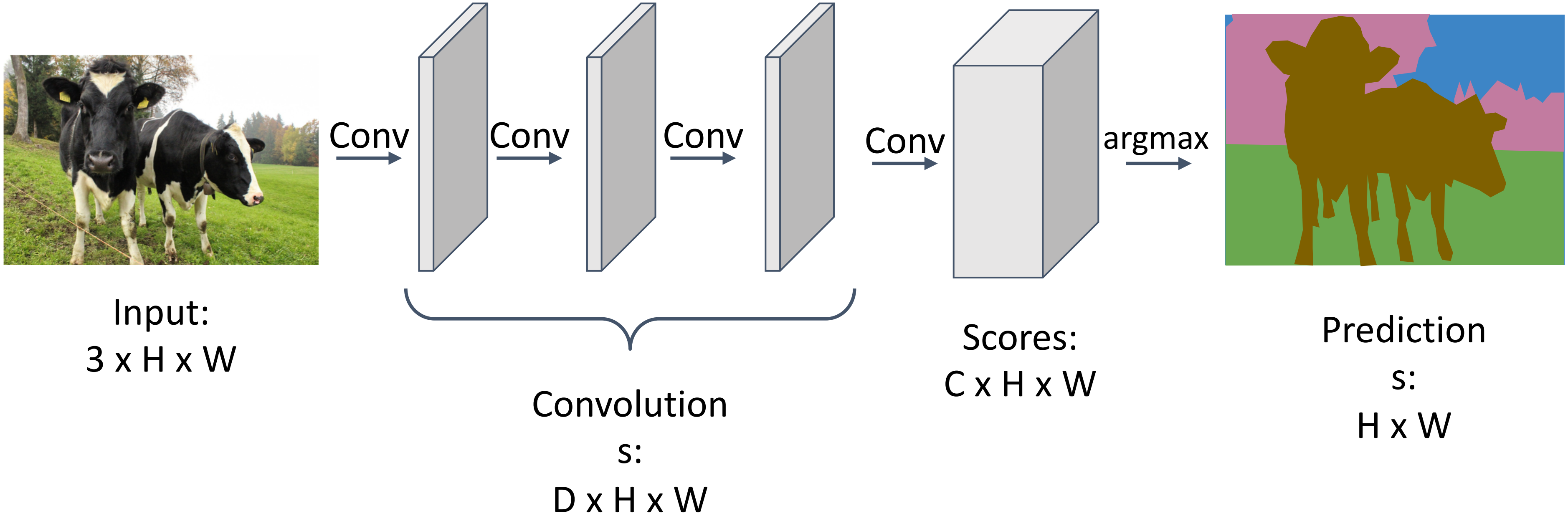
Grass

Problem: Very inefficient!
Not reusing shared features between overlapping patches

Farabet et al, "Learning Hierarchical Features for Scene Labeling," TPAMI 2013
Pinheiro and Collobert, "Recurrent Convolutional Neural Networks for Scene Labeling", ICML 2014

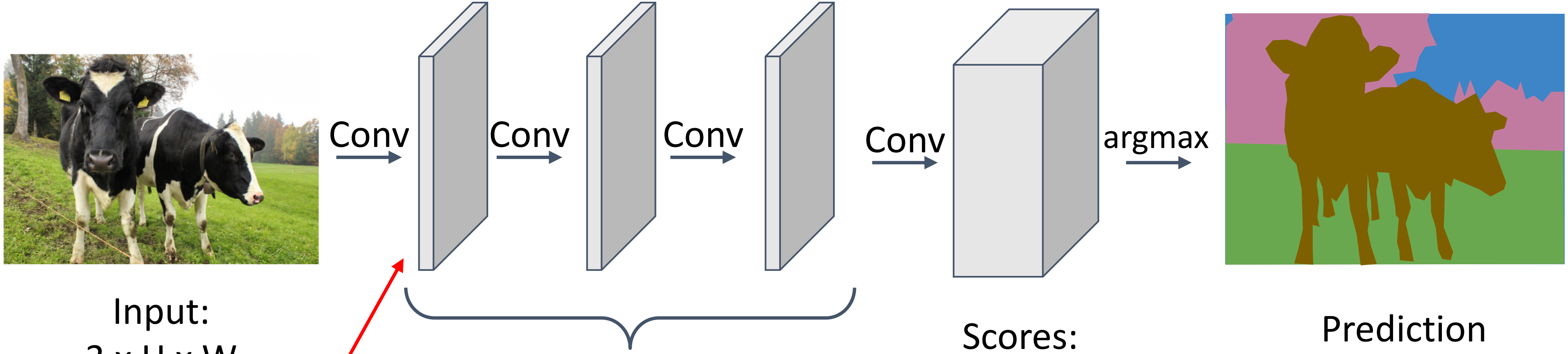
Semantic Segmentation Idea: Fully Convolutional

Design a network as a bunch of convolutional layers to make predictions for pixels all at once!



Semantic Segmentation Idea: Fully Convolutional

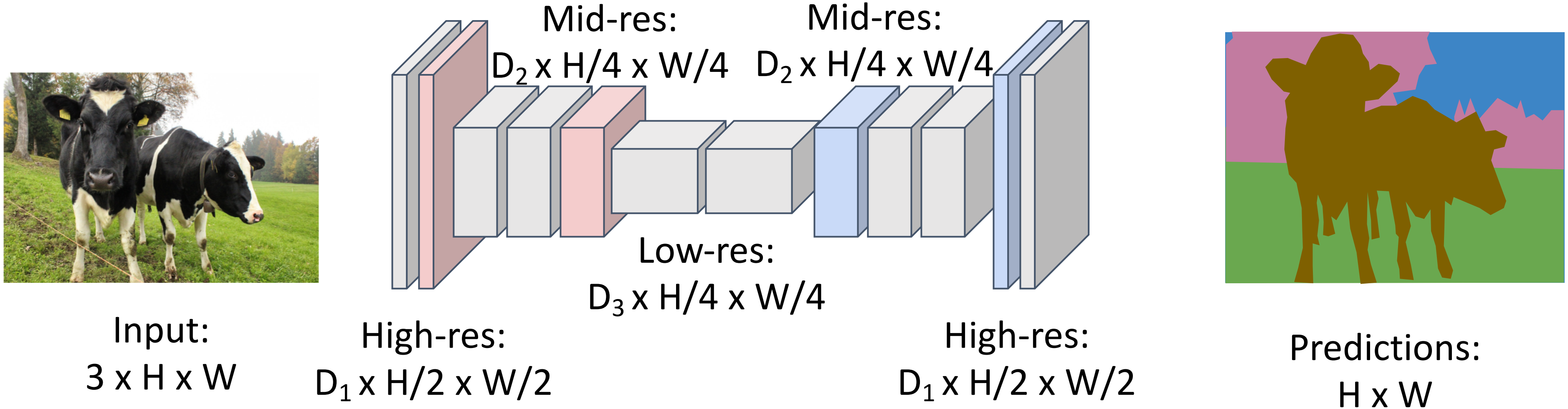
Design a network as a bunch of convolutional
layers to make predictions for pixels all at once!



Problem: convolutions at
original image resolution
will be very expensive ...

Semantic Segmentation Idea: Fully Convolutional

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!



Long, Shelhamer, and Darrell, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015
Noh et al, "Learning Deconvolution Network for Semantic Segmentation", ICCV 2015

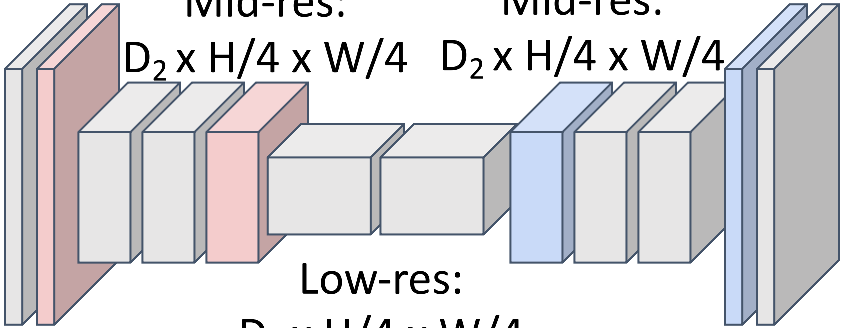
Semantic Segmentation Idea: Fully Convolutional

Design network as a bunch of convolutional layers, with **downsampling** and **upsampling** inside the network!

Downsampling: Pooling, strided convolution

Upsampling: Unpooling or strided transpose convolution

Input:
3 x H x W



Predictions:
H x W

Long, Shelhamer, and Darrell, “Fully Convolutional Networks for Semantic Segmentation”, CVPR 2015
Noh et al, “Learning Deconvolution Network for Semantic Segmentation”, ICCV 2015

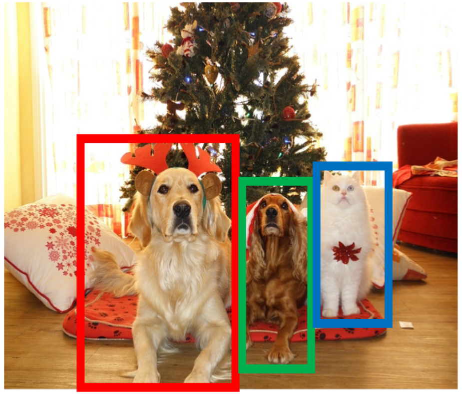
More Computer Vision Tasks

2D Semantic Segmentation

GRASS, CAT, TREE, SKY

Object categories + 2D segments

2D Object Detection



DOG, DOG, CAT

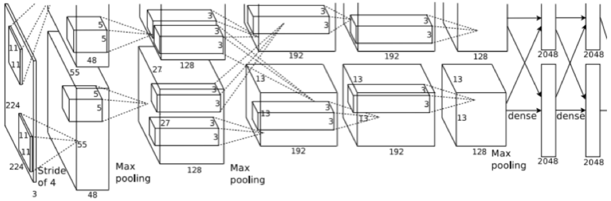
Object categories + 2D bounding boxes

3D Classificaion & Segmentation

Object categories + 3D segments

Classification + Localization

This image is CC0 public domain



Vector:
4096

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

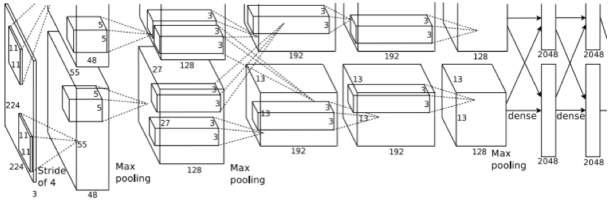
Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

Treat localization as a regression problem!

Classification + Localization

This image is CC0 public domain



Vector:
4096

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label:
Cat

Softmax Loss

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

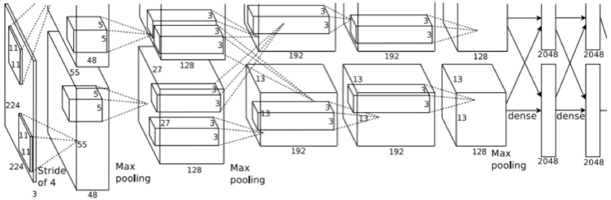
L2 Loss

Correct box:
(x’, y’, w’, h’)

Treat localization as a regression problem!

Classification + Localization

This image is CC0 public domain



Vector: 4096

Fully Connected: 4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label: Cat

Softmax Loss

Multitask Loss

+ → **Loss**

Fully Connected: 4096 to 4

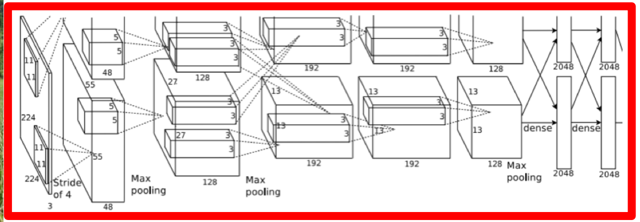
Box Coordinates (x, y, w, h)

L2 Loss

Correct box: (x’, y’, w’, h’)

Treat localization as a regression problem!

Classification + Localization



This image is CC0 public domain

Often pretrained on ImageNet
(Transfer learning)

Vector:
4096

Fully Connected:
4096 to 1000

Class Scores
Cat: 0.9
Dog: 0.05
Car: 0.01
...

Correct label:
Cat

Softmax Loss

Fully Connected:
4096 to 4

Box Coordinates
(x, y, w, h)

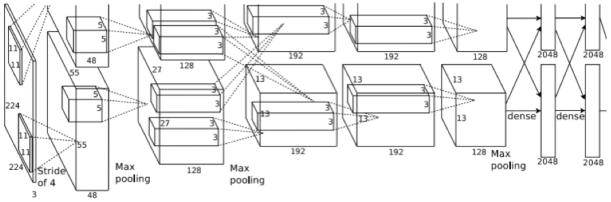
L2 Loss

Correct box:
(x’, y’, w’, h’)

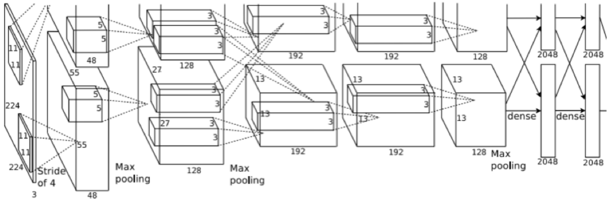
+ → **Loss**

Treat localization as a
regression problem!

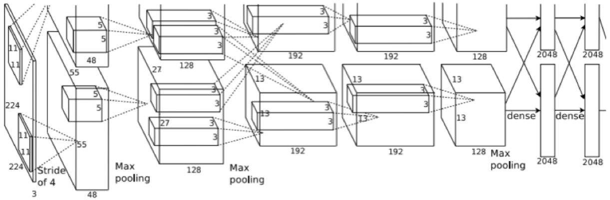
Object Detection as Regression?

CAT: (x, y, w, h)

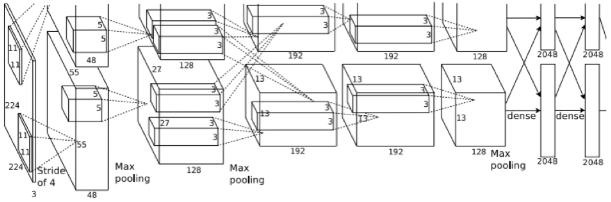
DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

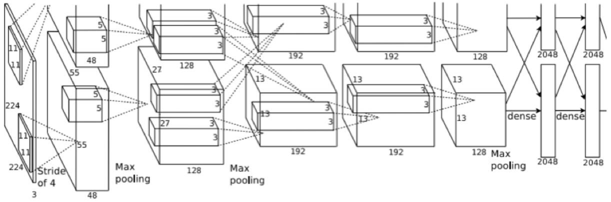
Object Detection as Regression?

Each image needs a different number of outputs!



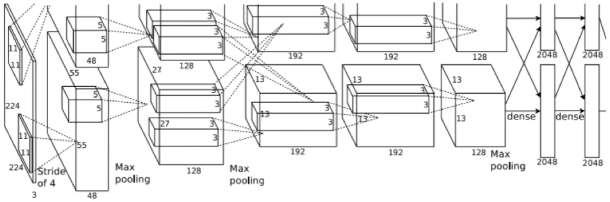
CAT: (x, y, w, h)

4 numbers



DOG: (x, y, w, h)
DOG: (x, y, w, h)
CAT: (x, y, w, h)

16 numbers

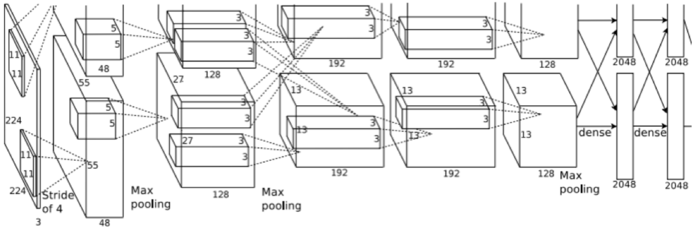


DUCK: (x, y, w, h)
DUCK: (x, y, w, h)
....

Many numbers!

Object Detection as Classification: Sliding Window

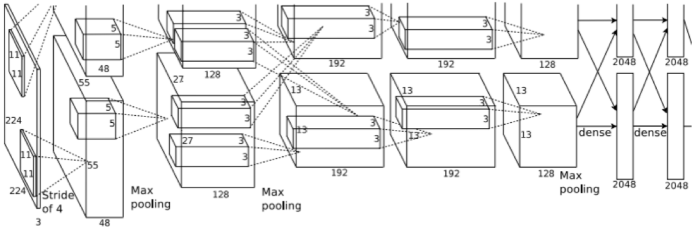
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



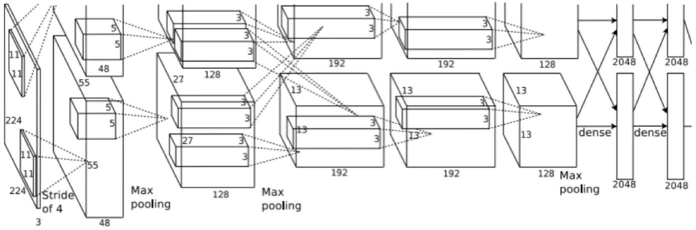
Dog? YES

Cat? NO

Background? NO

Object Detection as Classification: Sliding Window

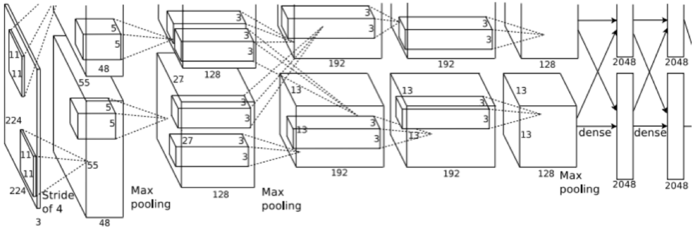
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? YES
Cat? NO
Background? NO

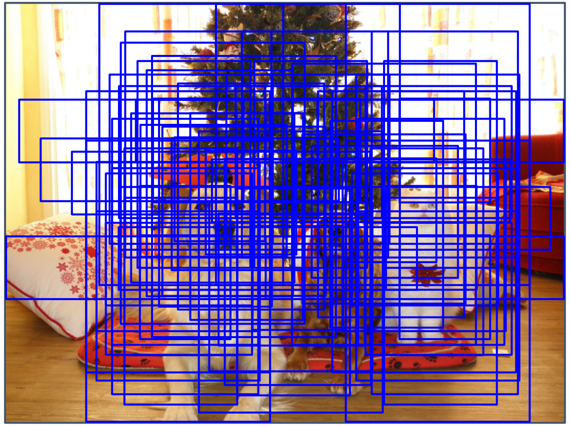
Object Detection as Classification: Sliding Window

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

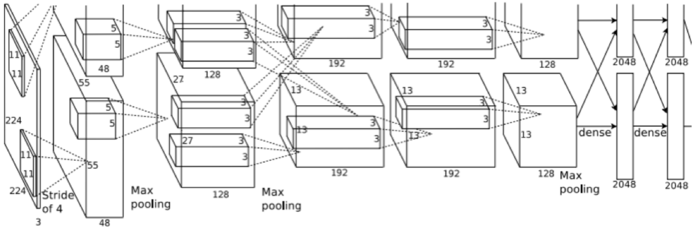


Dog? NO
Cat? YES
Background? NO

Object Detection as Classification: Sliding Window



Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

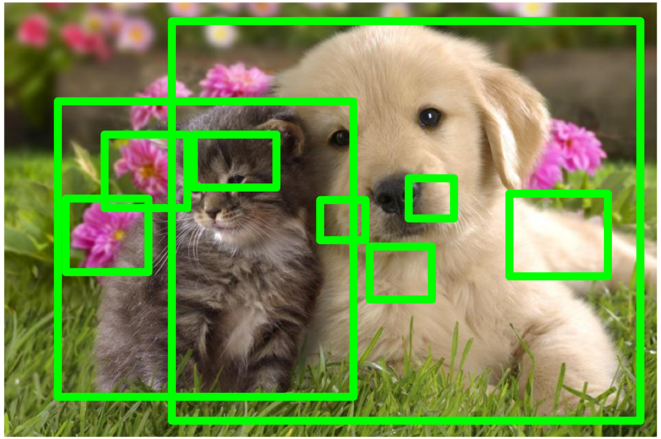


Dog? NO
Cat? YES
Background? NO

Problem: Need to apply CNN to huge number of locations, scales, and aspect ratios, very computationally expensive!

Region Proposals / Selective Search

- Find "blobby" image regions that are likely to contain objects
- Relatively fast to run; e.g. Selective Search gives 2000 region proposals in a few seconds on CPU



Alexe et al, "Measuring the objectness of image windows", TPAMI 2012
Uijlings et al, "Selective Search for Object Recognition", IJCV 2013
Cheng et al, "BING: Binarized normed gradients for objectness estimation at 300fps", CVPR 2014
Zitnick and Dollar, "Edge boxes: Locating object proposals from edges", ECCV 2014

R-CNN

Input image

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

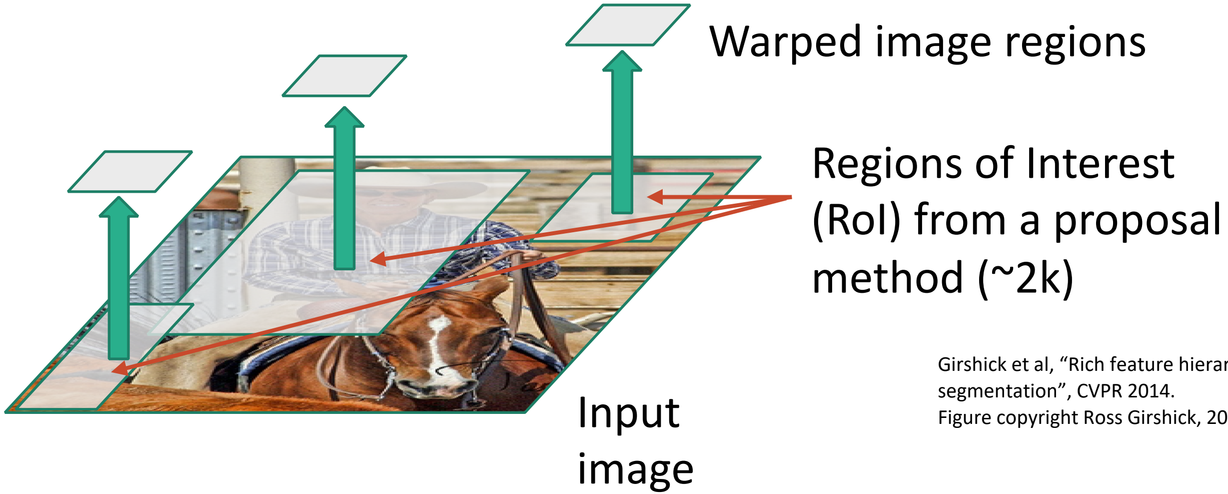
R-CNN

Regions of Interest (RoI) from a proposal method (~2k)

Input image

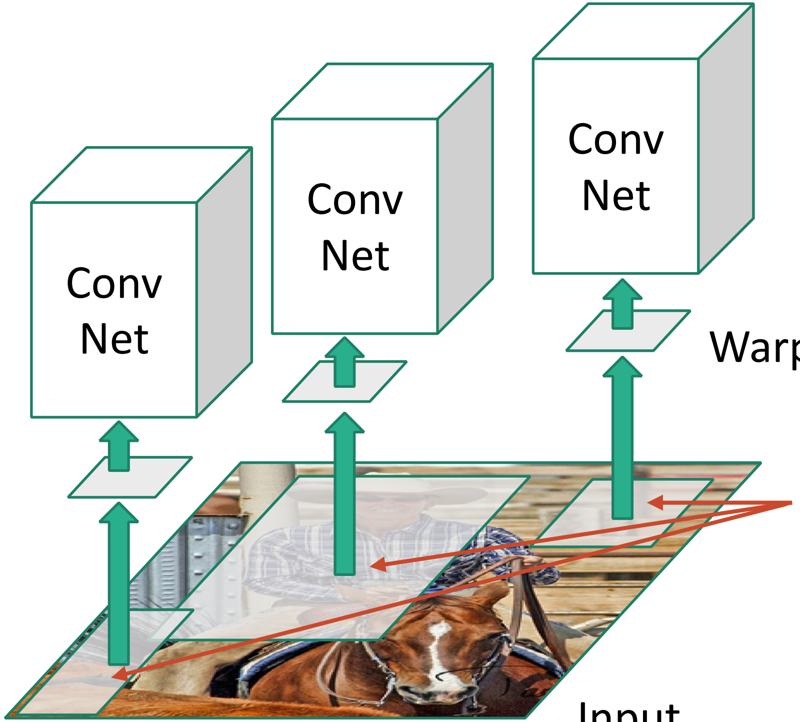
Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN



Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
Figure copyright Ross Girshick, 2015; source. Reproduced with permission.

R-CNN



Forward each region through ConvNet

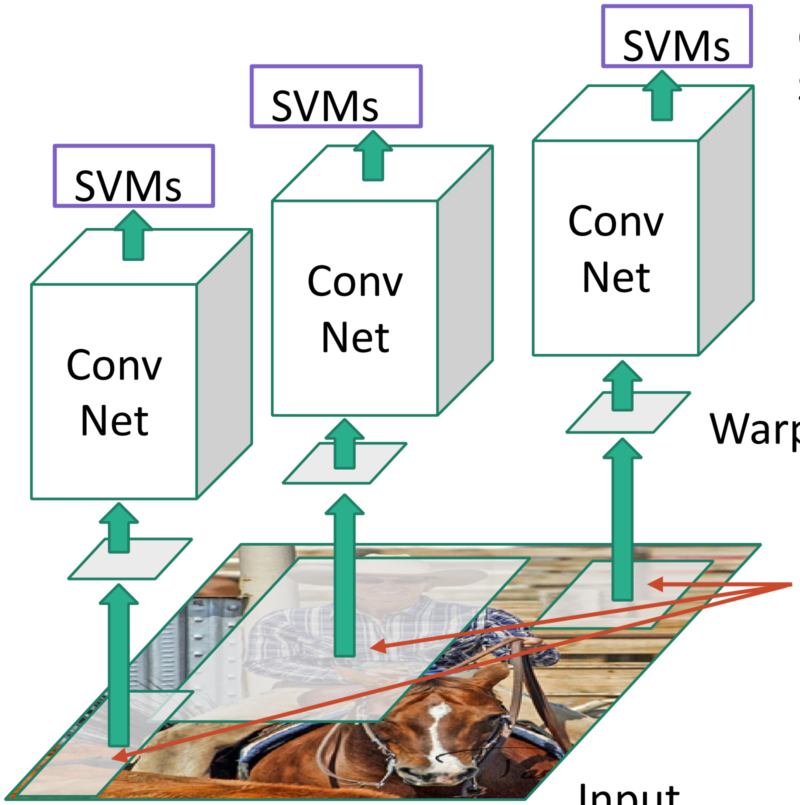
Warped image regions

Regions of Interest (RoI) from a proposal method (~2k)

Input image

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

R-CNN



Classify regions with SVMs

Forward each region through ConvNet

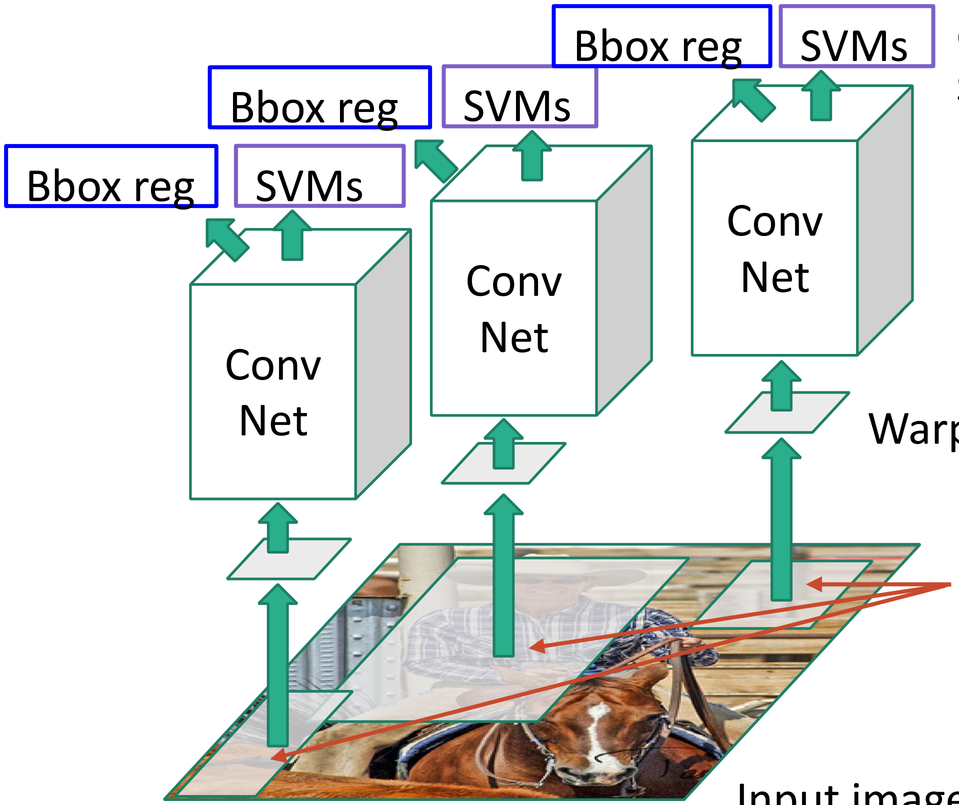
Warped image regions

Regions of Interest (RoI) from a proposal method (~2k)

Input image

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

R-CNN



Linear Regression for bounding box offsets

Classify regions with SVMs

Forward each region through ConvNet

Warped image regions

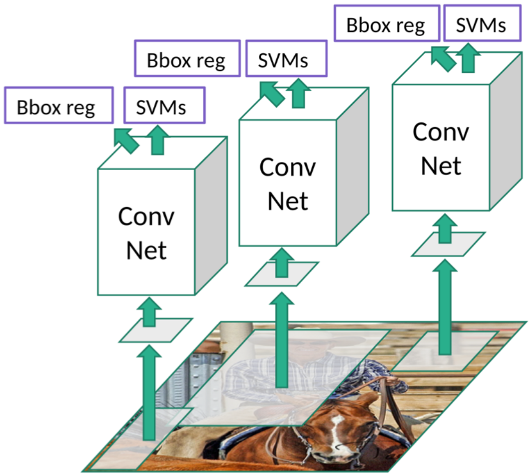
Regions of Interest (RoI) from a proposal method (~2k)

Input image

Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.

R-CNN: Problems

- Ad hoc training objectives
  - Fine-tune network with softmax classifier (log loss)
  - Train post-hoc linear SVMs (hinge loss)
  - Train post-hoc bounding-box regressions (least squares)
- Training is slow (84h), takes a lot of disk space
- Inference (detection) is slow
  - 47s / image with VGG16 [Simonyan & Zisserman. ICLR15]
  - Fixed by SPP-net [He et al. ECCV14]



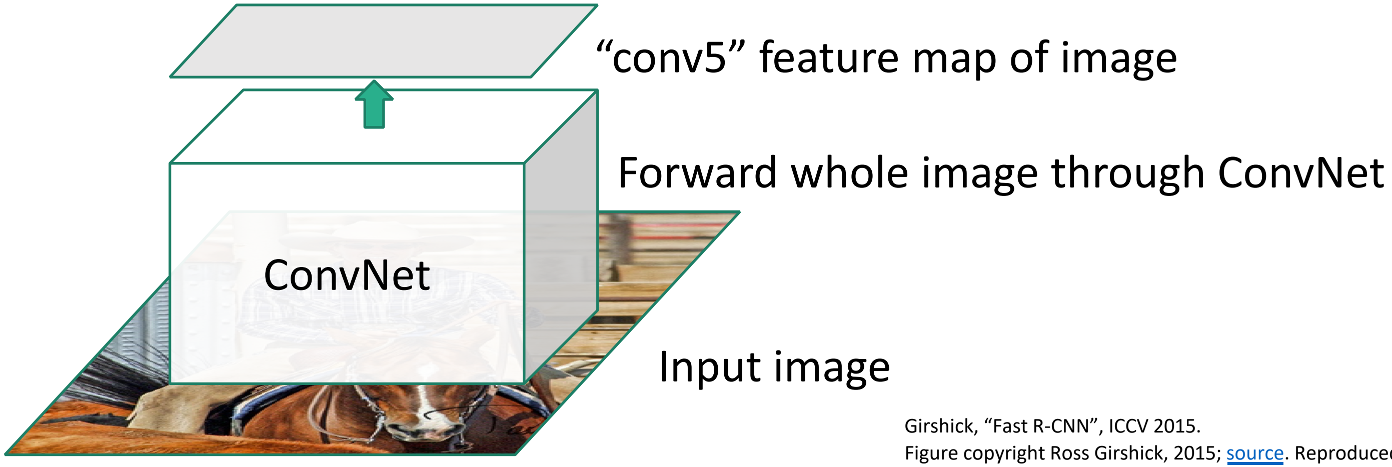
Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.

Fast R-CNN

Input image

Girshick, “Fast R-CNN”, ICCV 2015.

Fast R-CNN

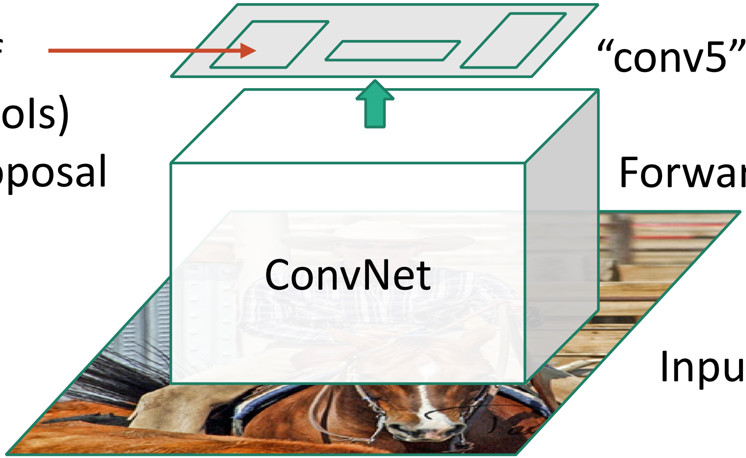


Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN

Regions of Interest (RoIs) from a proposal method

“conv5” feature map of image

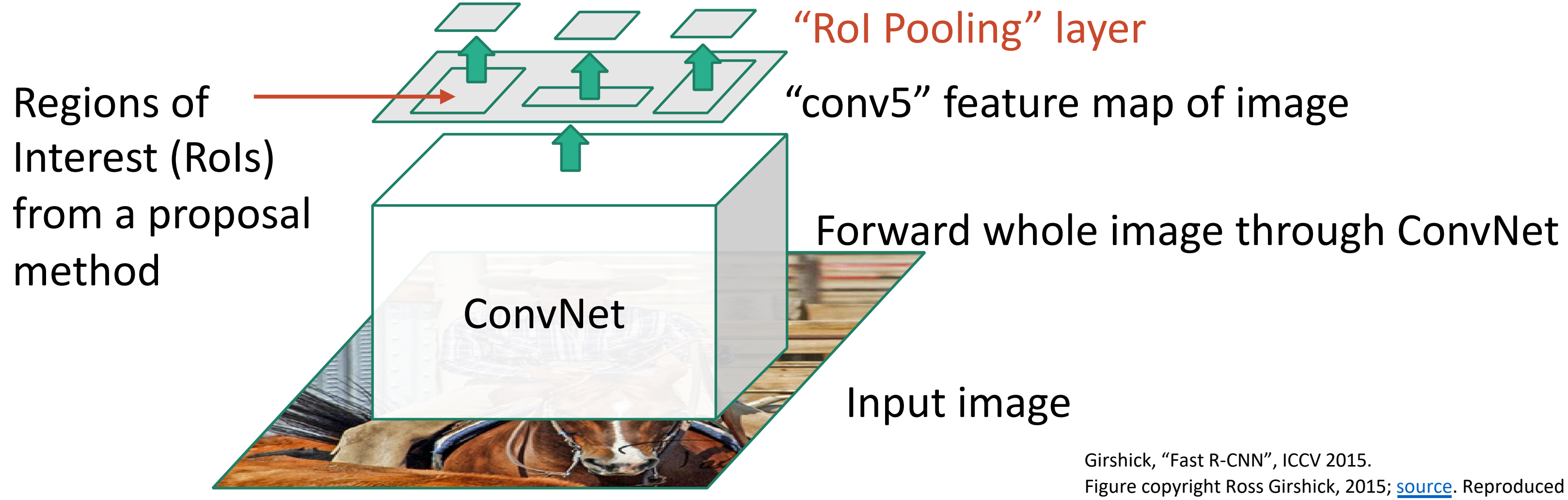


Forward whole image through ConvNet

Input image

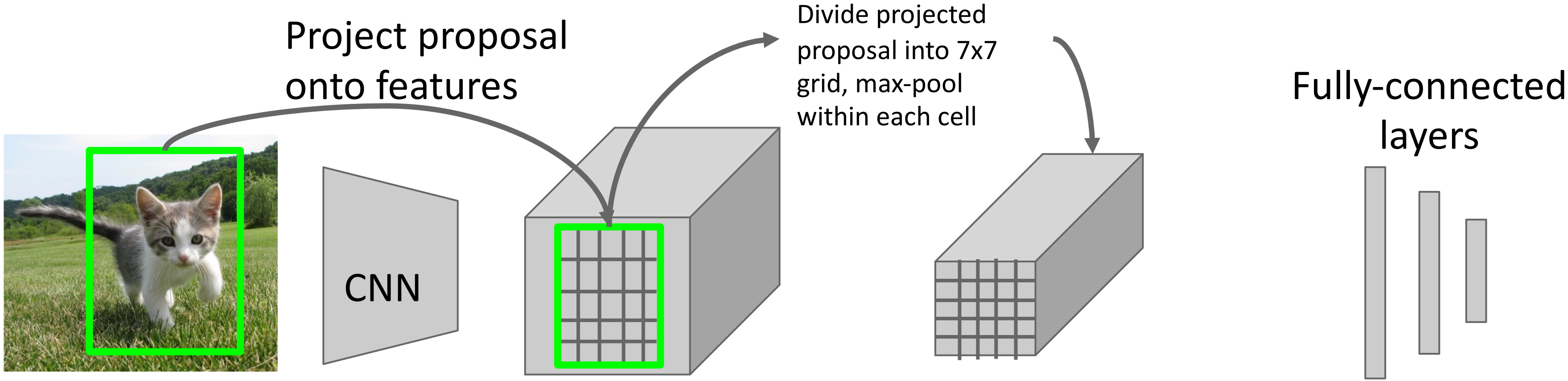
Girshick, “Fast R-CNN”, ICCV 2015.

Fast R-CNN



Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN: RoI Pooling



Hi-res input image:
3 x 640 x 480
with region proposal

Hi-res conv features:
512 x 20 x 15;

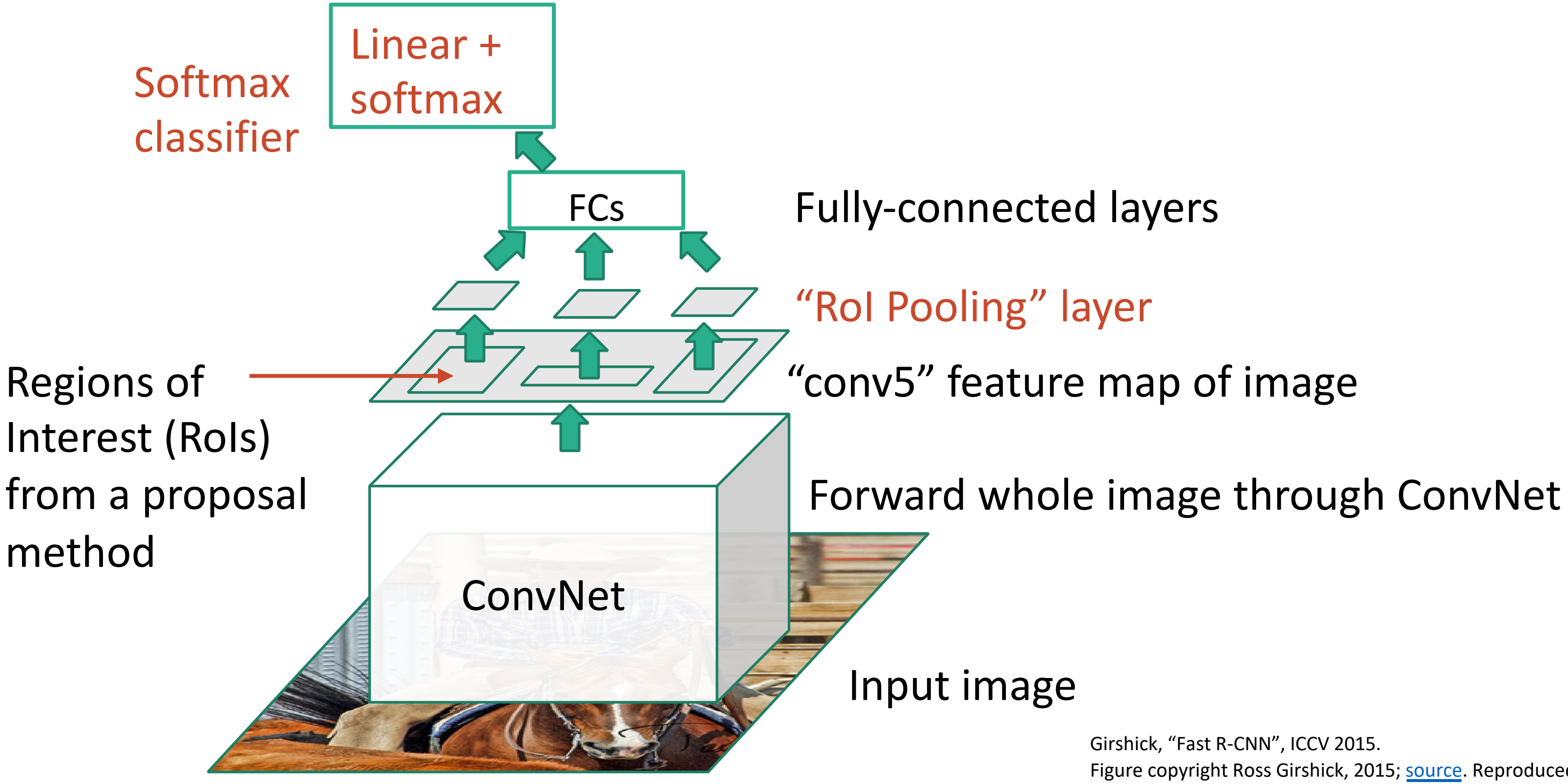
Projected region proposal is e.g.
512 x 18 x 8
(varies per proposal)

RoI conv features:
512 x 7 x 7
for region proposal

Fully-connected layers expect low-res conv features:
512 x 7 x 7

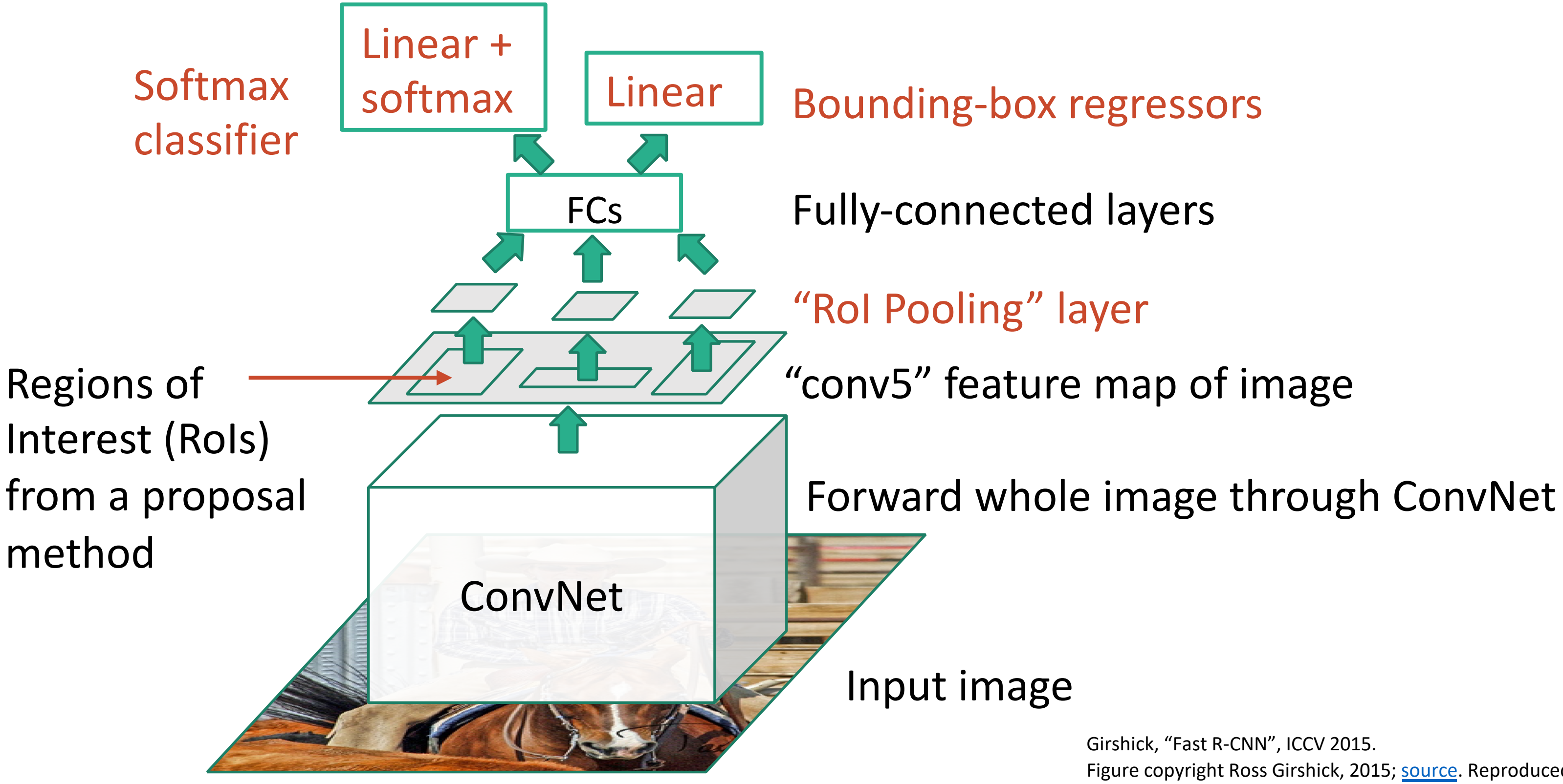
Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN



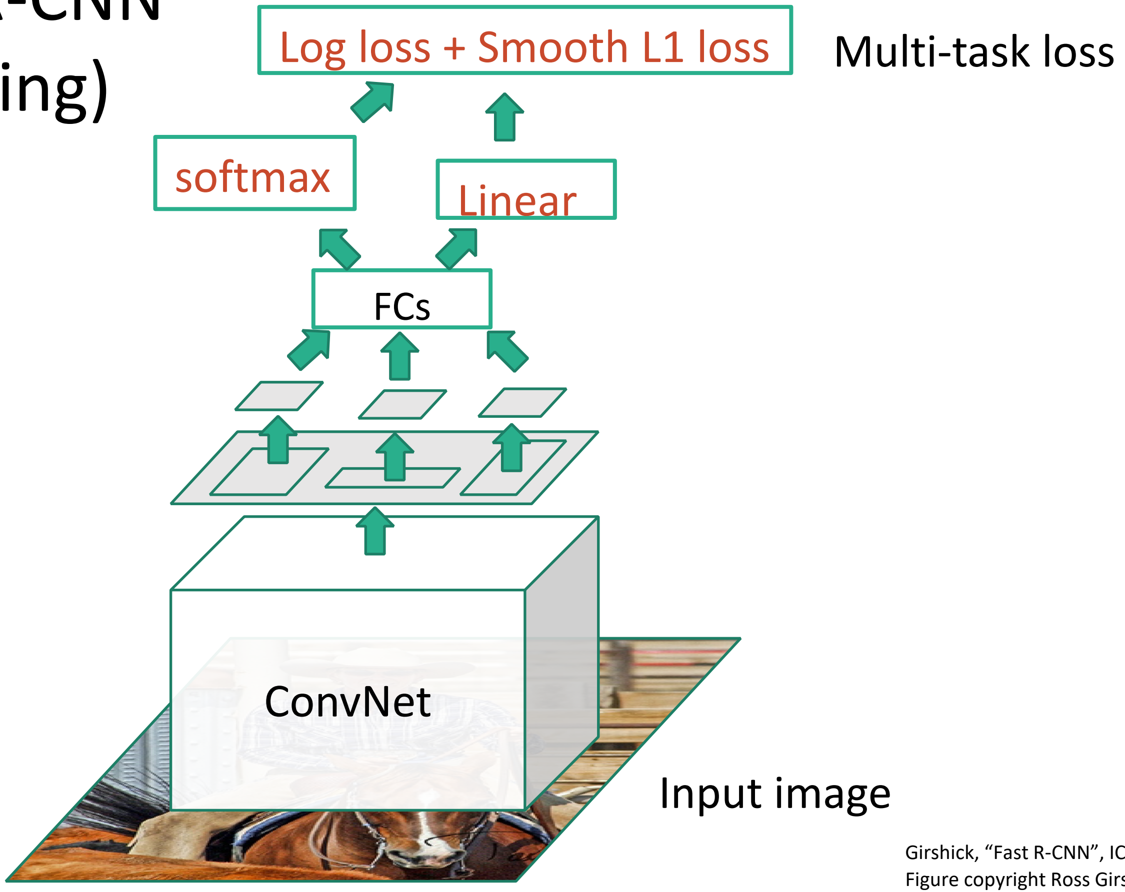
Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN



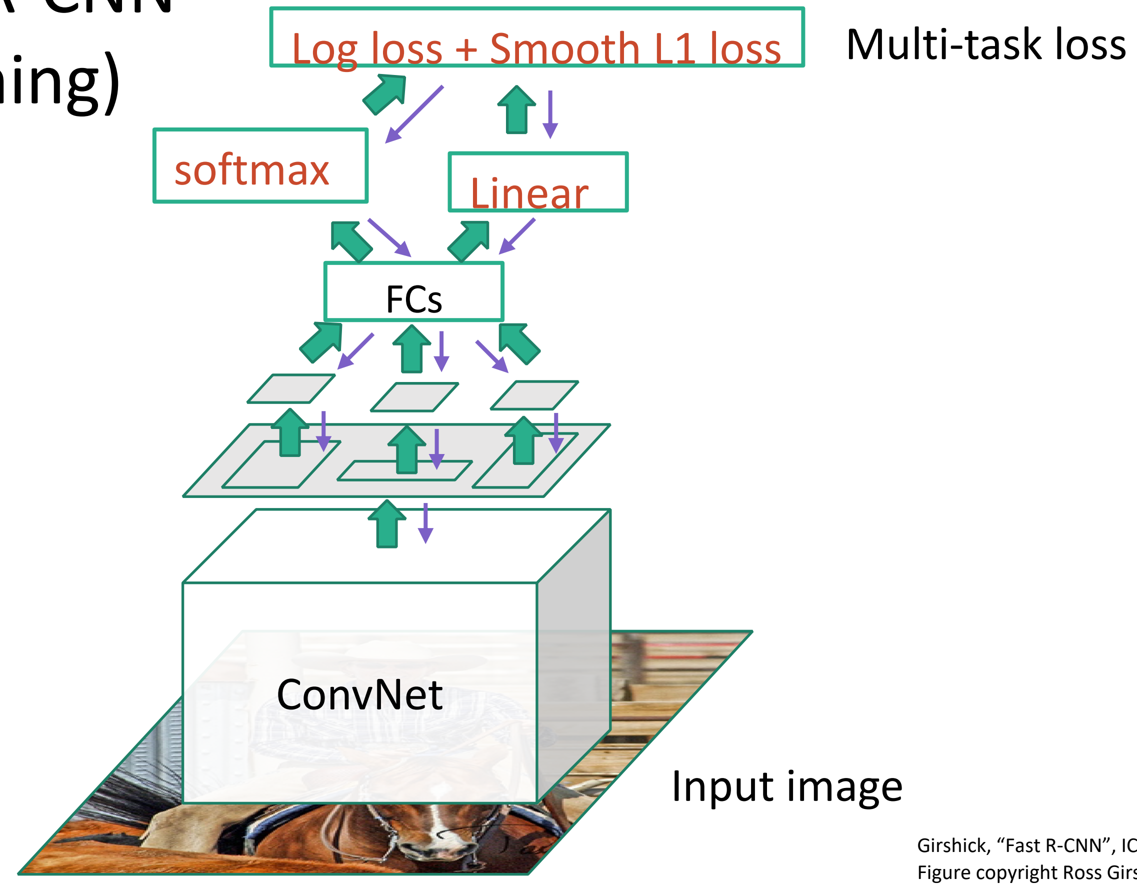
Girshick, "Fast R-CNN", ICCV 2015.

Fast R-CNN (Training)



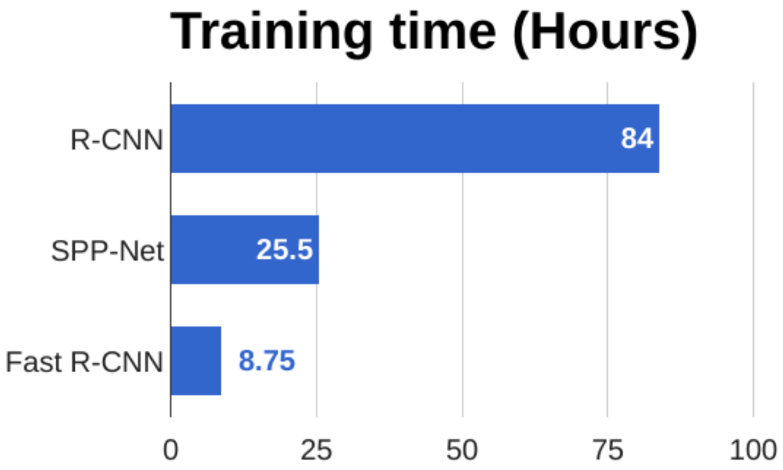
Girshick, "Fast R-CNN", ICCV 2015.

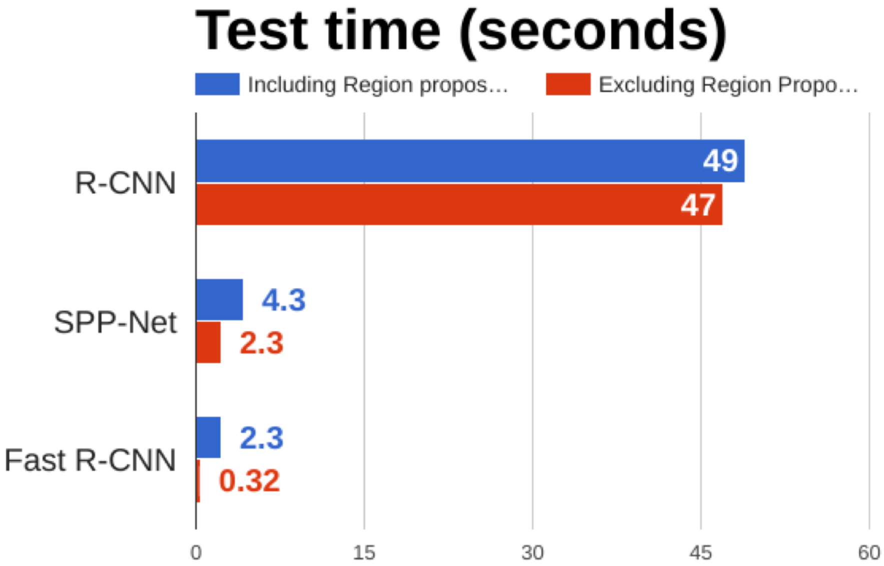
Fast R-CNN (Training)



Girshick, "Fast R-CNN", ICCV 2015.

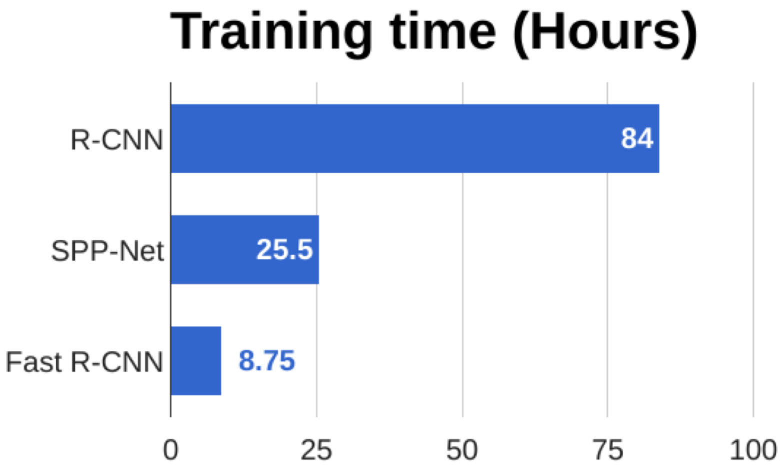
R-CNN vs SPP vs Fast R-CNN

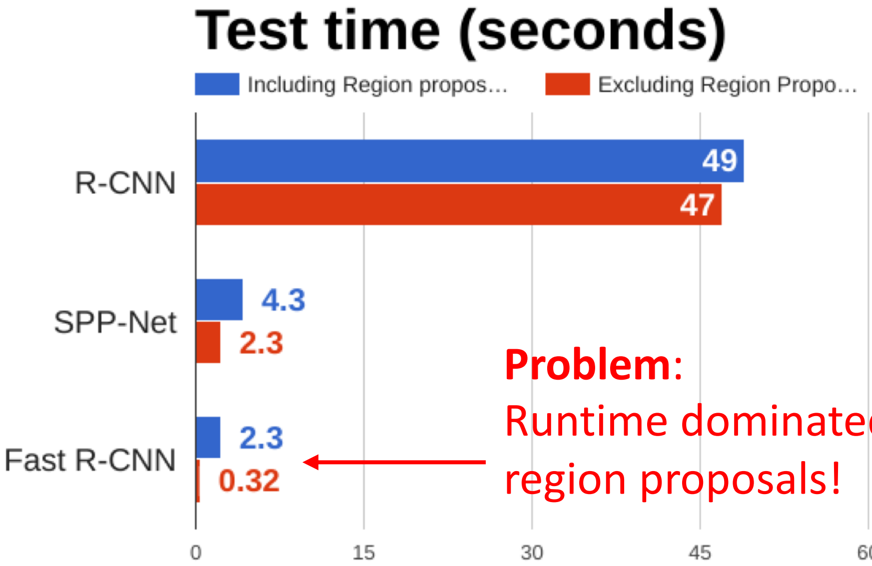




Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

R-CNN vs SPP vs Fast R-CNN





Problem:
Runtime dominated by region proposals!

Girshick et al, "Rich feature hierarchies for accurate object detection and semantic segmentation", CVPR 2014.
He et al, "Spatial pyramid pooling in deep convolutional networks for visual recognition", ECCV 2014
Girshick, "Fast R-CNN", ICCV 2015

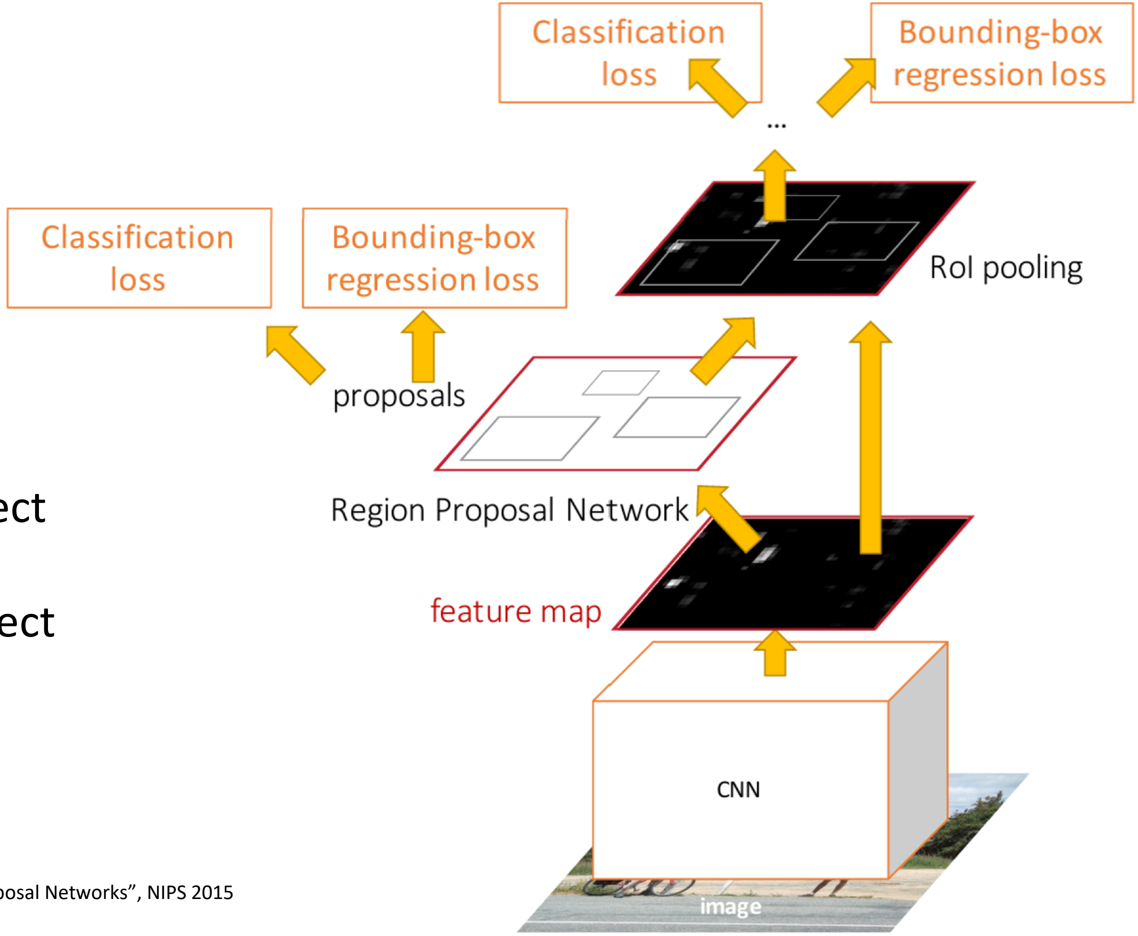
Faster R-CNN:

Make CNN do proposals!

Insert **Region Proposal Network (RPN)** to predict proposals from features

Jointly train with 4 losses:

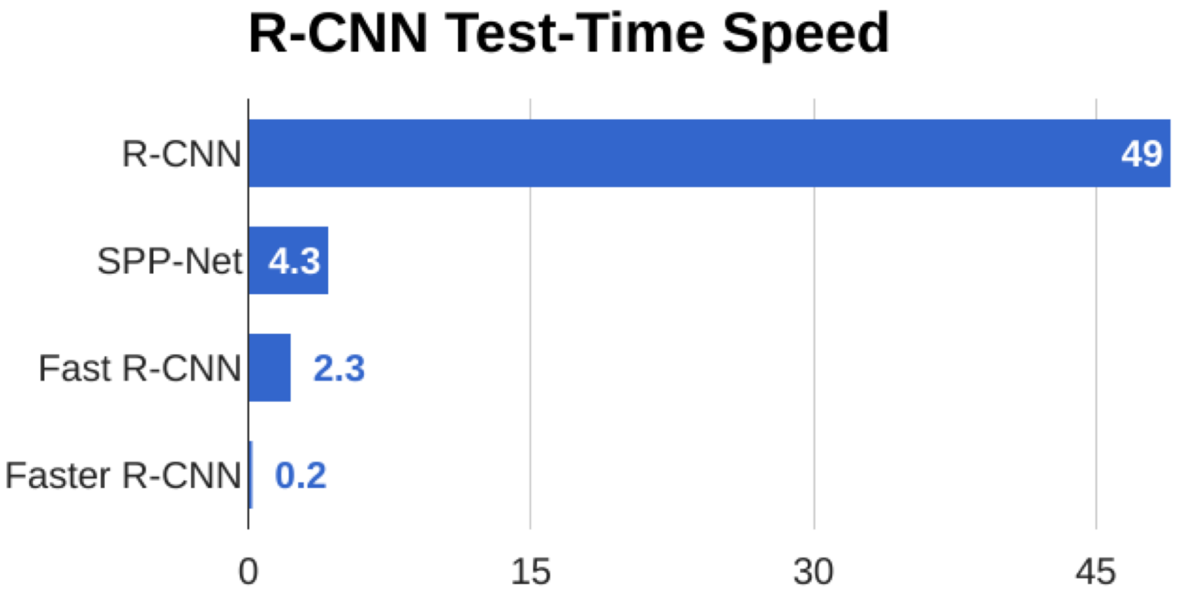
1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates



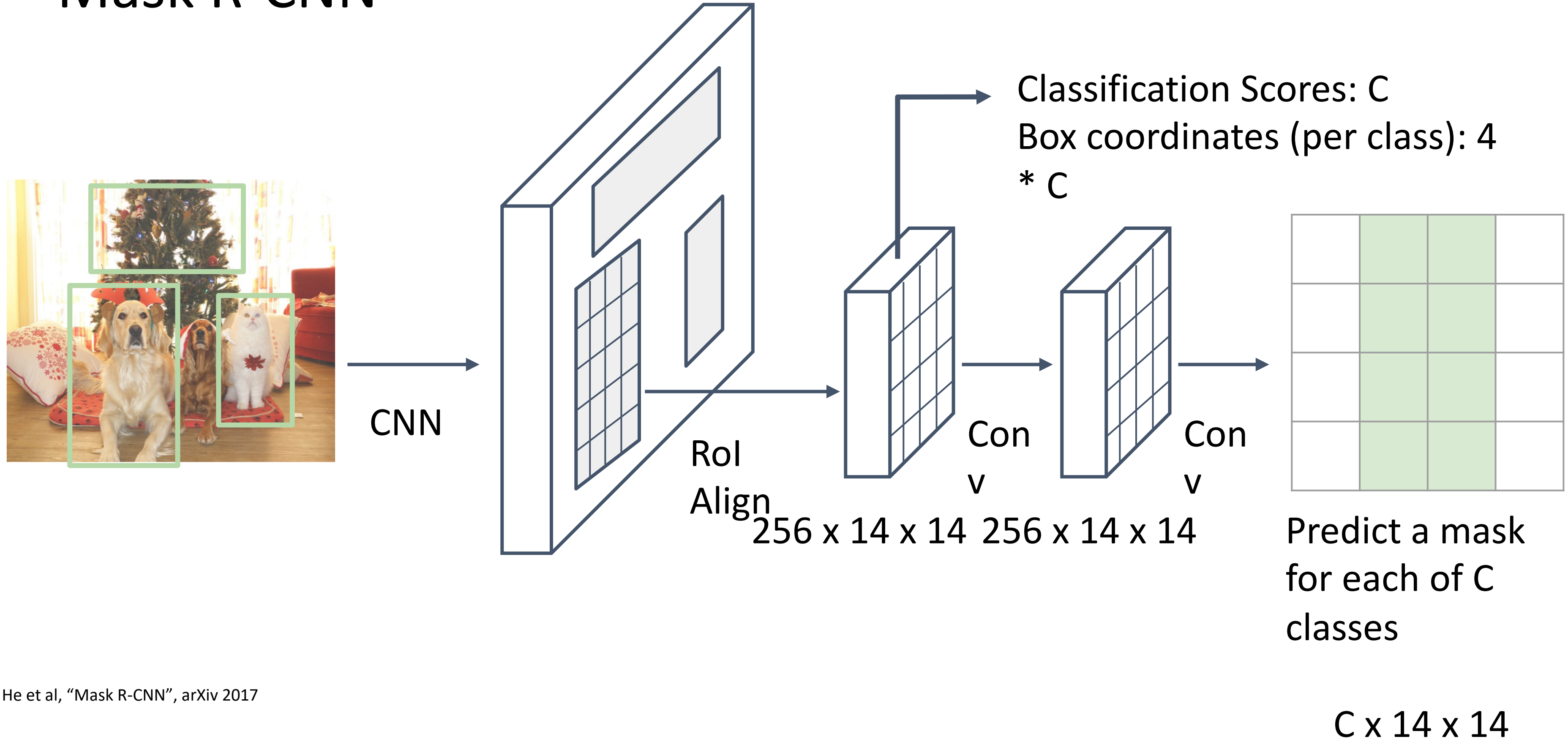
Ren et al, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, NIPS 2015

Faster R-CNN:

Make CNN do proposals!

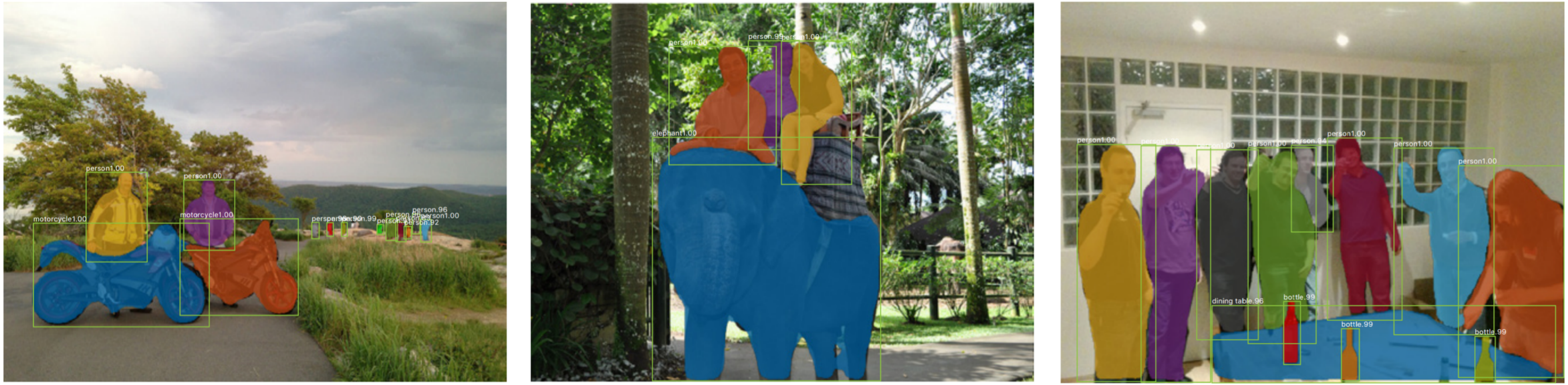


Mask R-CNN



He et al, "Mask R-CNN", arXiv 2017

Mask R-CNN: Very Good Results!



He et al, “Mask R-CNN”, arXiv 2017

Mask R-CNN: Also predict pose

He et al, “Mask R-CNN”, arXiv 2017

Object Detection: Lots of variables ...

Base Network
VGG16
ResNet-101
Inception V2
Inception V3
Inception ResNet
MobileNet

Object Detection architecture
Faster R-CNN
R-FCN
SSD

Image Size
 # Region Proposals
…

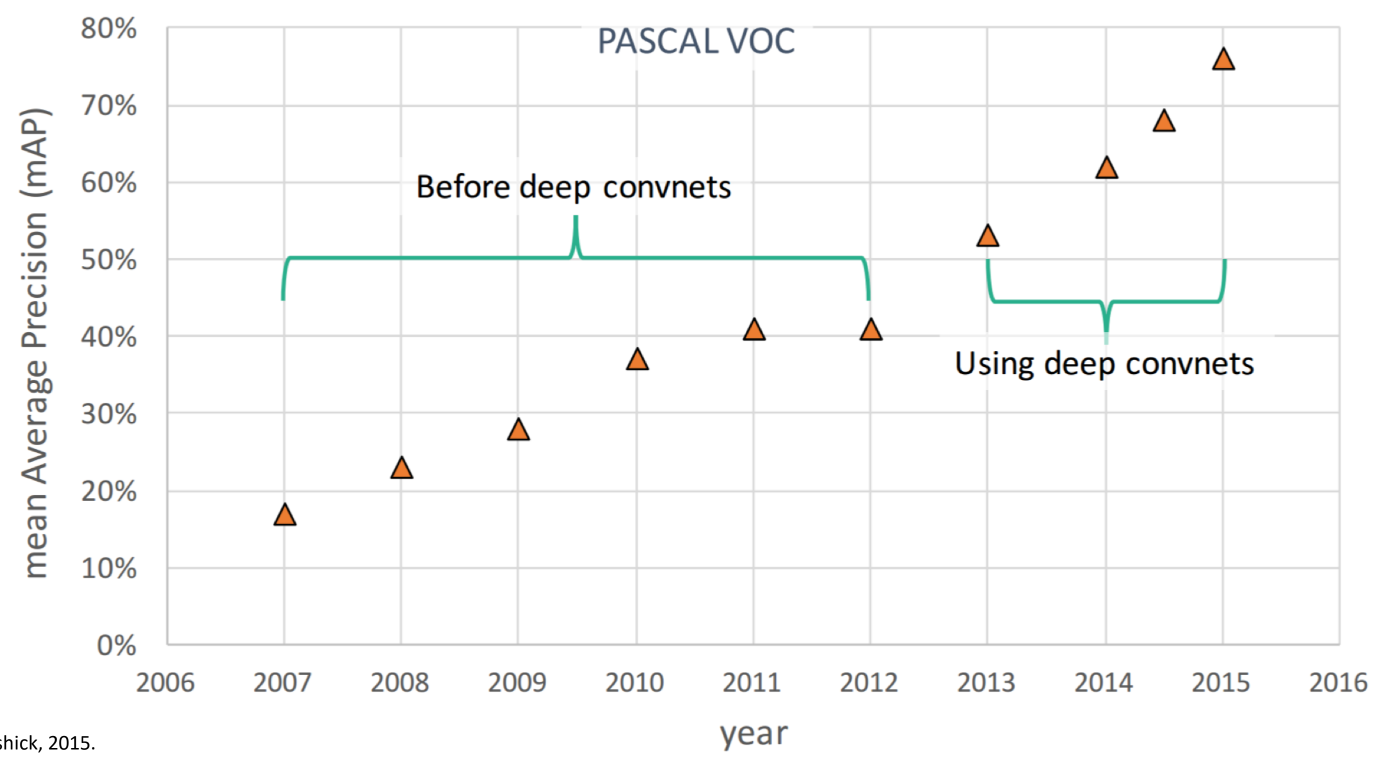
Takeaways
Faster R-CNN is slower but more accurate

SSD is much faster but not as accurate

Huang et al, "Speed/accuracy trade-offs for modern convolutional object detectors", CVPR 2017

R-FCN: Dai et al, "R-FCN: Object Detection via Region-based Fully Convolutional Networks", NIPS 2016
Inception-V2: Ioffe and Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift", ICML 2015
Inception V3: Szegedy et al, "Rethinking the Inception Architecture for Computer Vision", arXiv 2016
Inception ResNet: Szegedy et al, "Inception-V4, Inception-ResNet and the Impact of Residual Connections on Learning", arXiv 2016
MobileNet: Howard et al, "Efficient Convolutional Neural Networks for Mobile Vision Applications", arXiv 2017

Object Detection: Impact of Deep Learning



Open Source Frameworks

Lots of good implementations on GitHub!

TensorFlow Detection API:

https://github.com/tensorflow/models/tree/master/research/object_detection
Faster RCNN, SSD, RFCN, Mask R-CNN

Caffe2 Detectron:
https://github.com/facebookresearch/Detectron
Mask R-CNN, RetinaNet, Faster R-CNN, RPN, Fast R-CNN, R-FCN

Finetune on your own dataset with pre-trained models

More Computer Vision Tasks

2D Semantic Segmentation

GRASS, CAT, TREE, SKY

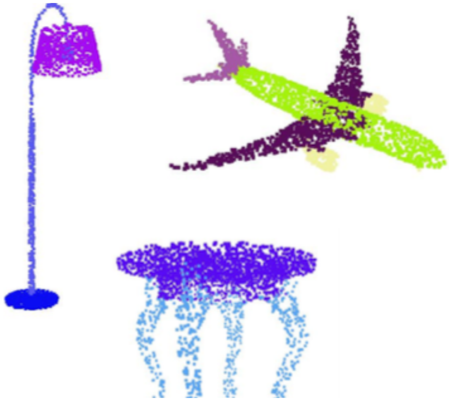
Object categories + 2D segments

2D Object Detection

DOG, DOG, CAT

Object categories + 2D bounding boxes

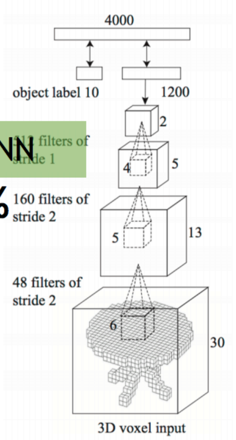
3D Classificaion & Segmentation



Object categories + 3D segments

3D CNNs on Voxelized Data

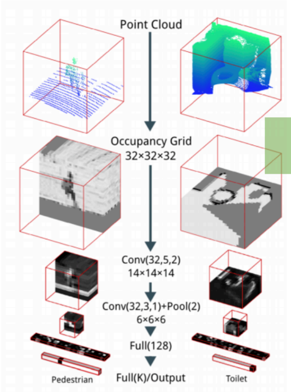
3DShapeNets from Princeton
CVPR 2015



3D CNN

77.3%

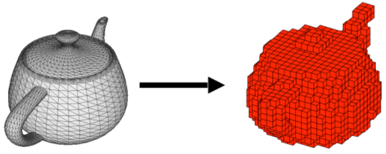
VoxNet from CMU Robotics
IEEE/RSJ 2015



3D CNN

83.0%

Information loss in voxelization



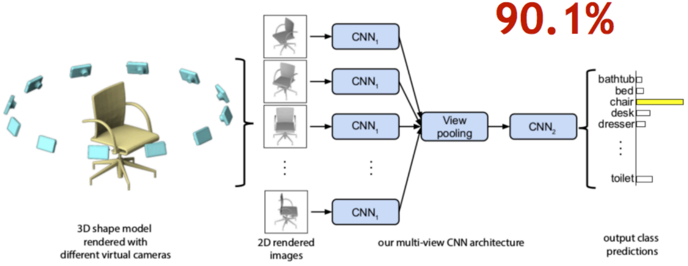
CAD model

Occupancy Grid
30x30x30

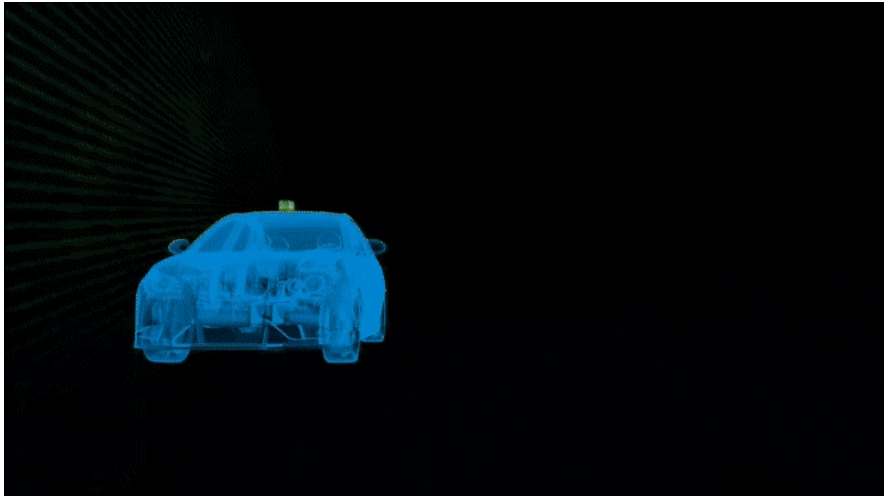
Rendering +
2D CNN

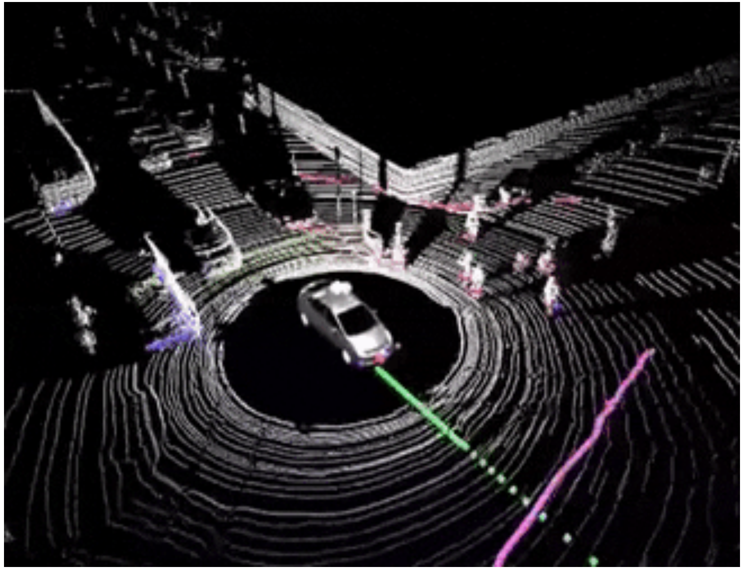
90.1%

MVCNN from UMass
ICCV 2015



Unordered Point Set

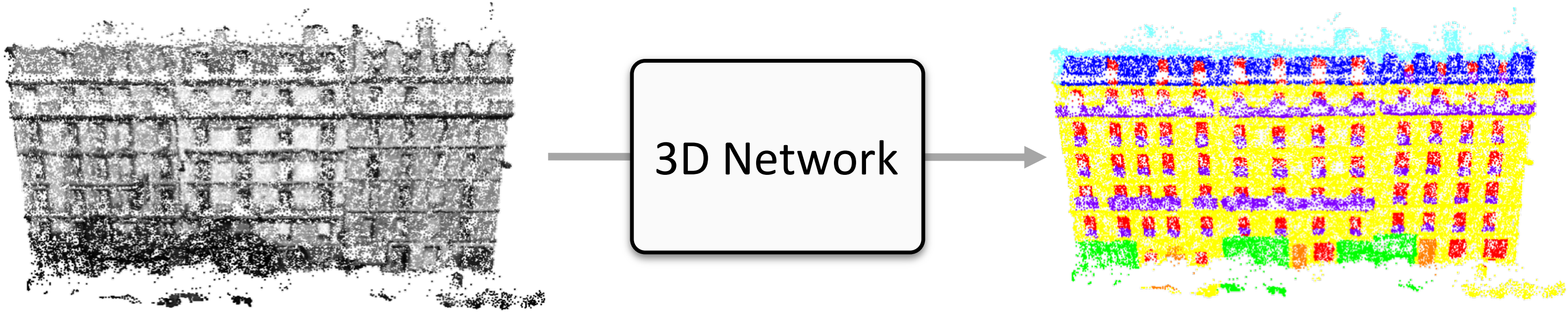




Lidar scan from autonomous vehicles

(Deep) Learning on 3D point sets

PointNet [Qi et al., CVPR 2017]

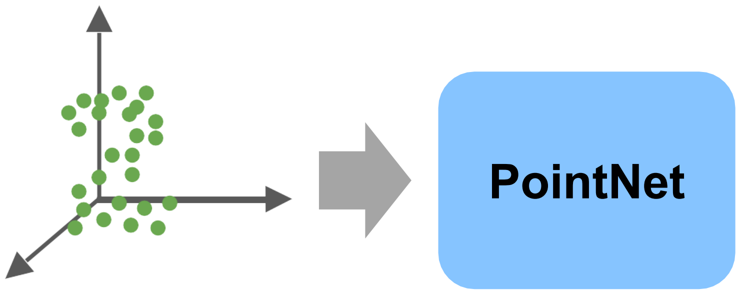


Input 3D point cloud

3D prediction

PointNet

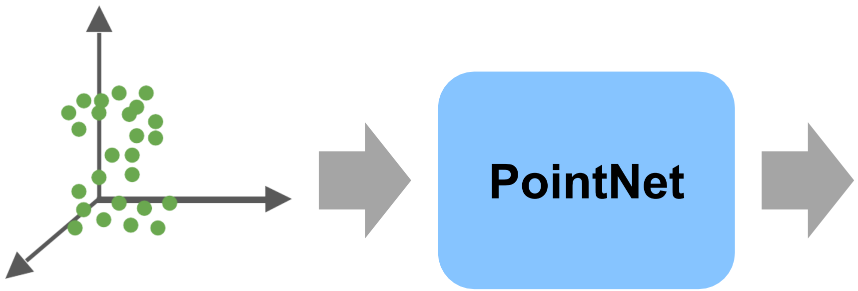
End-to-end learning for **scattered, unordered** point data



PointNet

End-to-end learning for **scattered, unordered** point data

Unified framework for various tasks



Object Classification

Object Part Segmentation

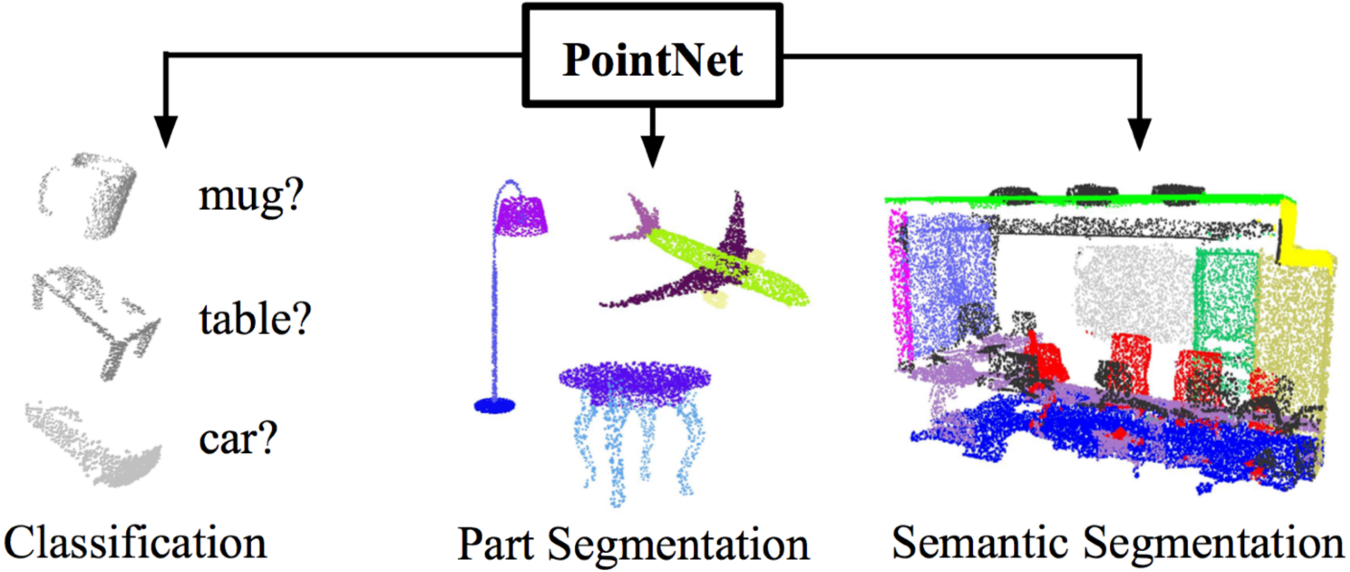
Semantic Scene Parsing

...

PointNet

End-to-end learning for **scattered, unordered** point data

Unified framework for various tasks



Challenges

Unordered point set as input

Model needs to be invariant to N! permutations.

Invariance under geometric transformations

Point cloud rotations should not alter classification results.

Challenges

Unordered point set as input

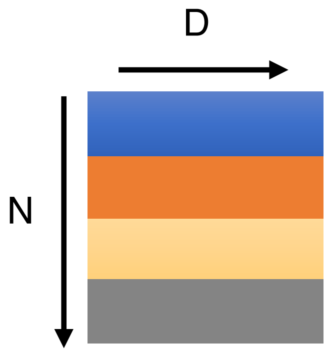
Model needs to be invariant to N! permutations.

Invariance under geometric transformations

Point cloud rotations should not alter classification results.

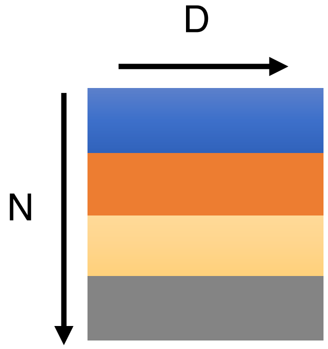
Unordered Input

Point cloud: N **orderless** points, each represented by a D dim vector

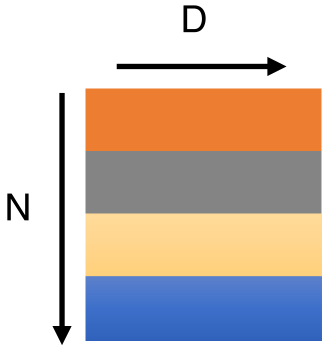


Unordered Input

Point cloud: N **orderless** points, each represented by a D dim vector

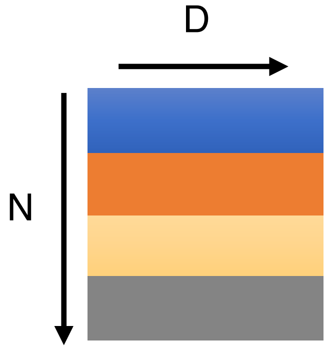


represents the same **set** as

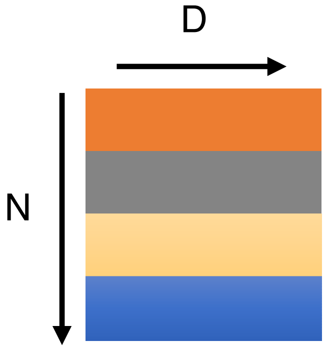


Unordered Input

Point cloud: N <u>**orderless**</u> points, each represented by a D dim vector



represents the same **set** as



Model needs to be invariant to N! permutations

Permutation Invariance: Symmetric Function

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Permutation Invariance: Symmetric Function

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

Examples:

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

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

…

Permutation Invariance: Symmetric Function

$$f(x_1, x_2, \ldots, x_n) \equiv f(x_{\pi_1}, x_{\pi_2}, \ldots, x_{\pi_n}), \quad x_i \in \mathbb{R}^D$$

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 family of symmetric functions by neural networks?

Permutation Invariance: Symmetric Function

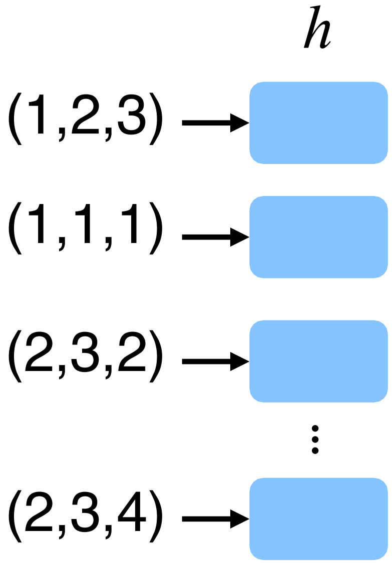
Observe:

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

Permutation Invariance: Symmetric Function

Observe:

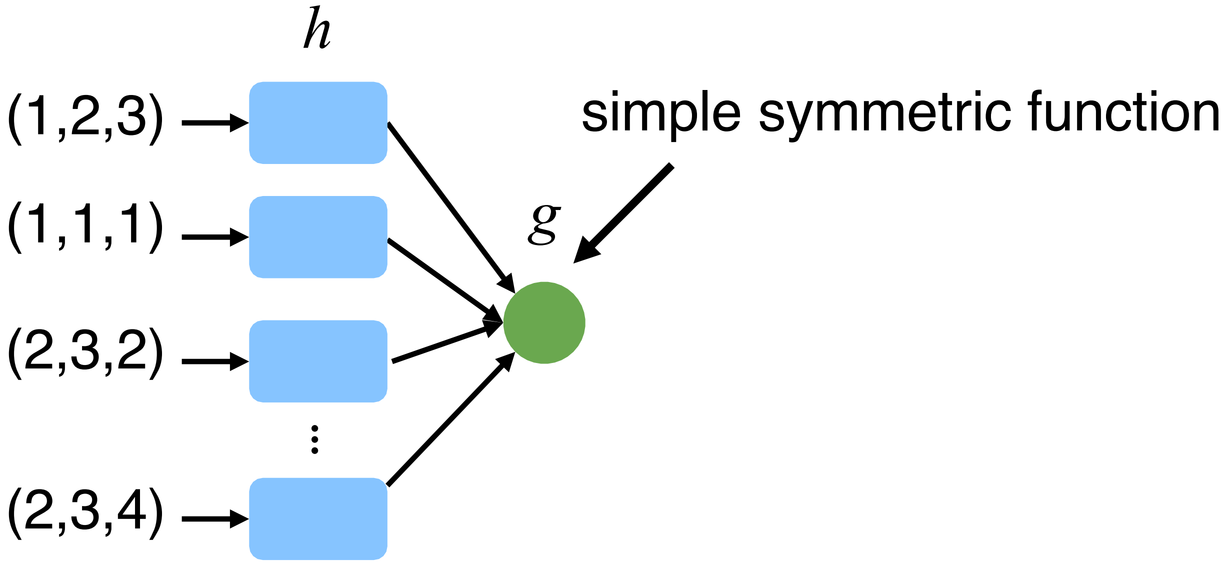
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Permutation Invariance: Symmetric Function

Observe:

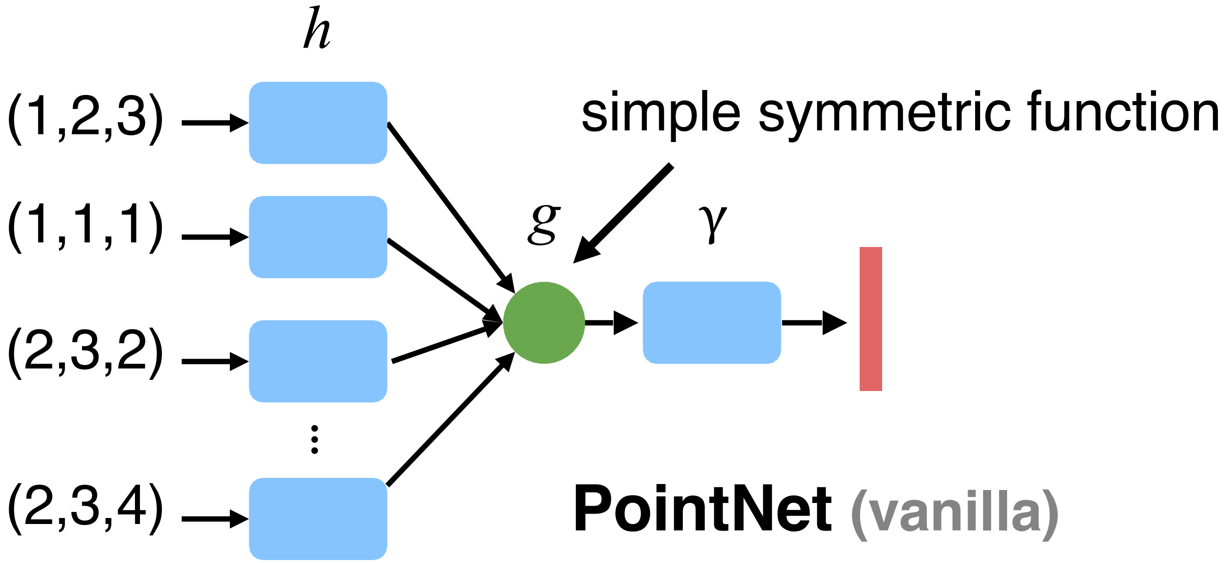
$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$ is symmetric if g is symmetric



Permutation Invariance: Symmetric Function

Observe:

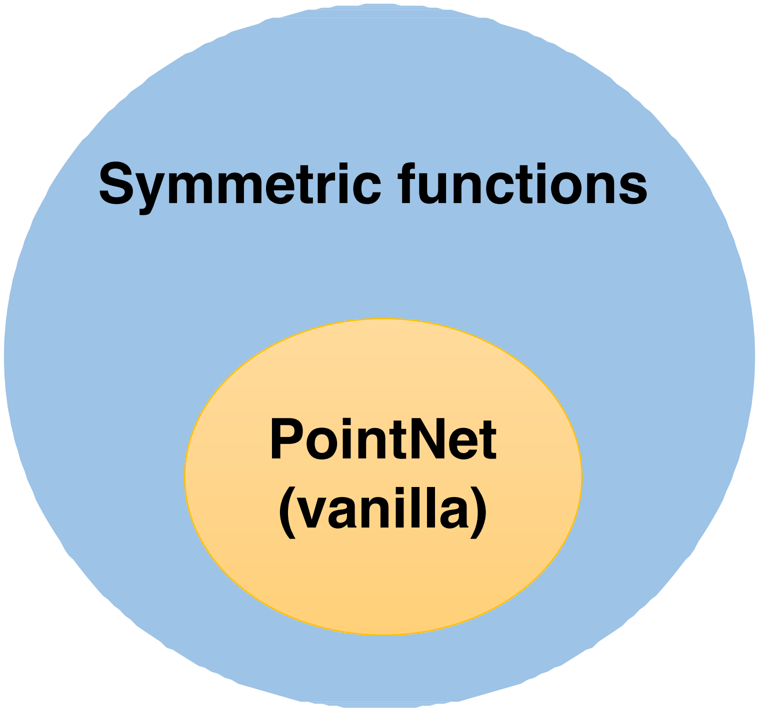
$$f(x_1, x_2, \ldots, x_n) = \gamma \circ g(h(x_1), \ldots, h(x_n))$$ is symmetric if g is symmetric



PointNet (vanilla)

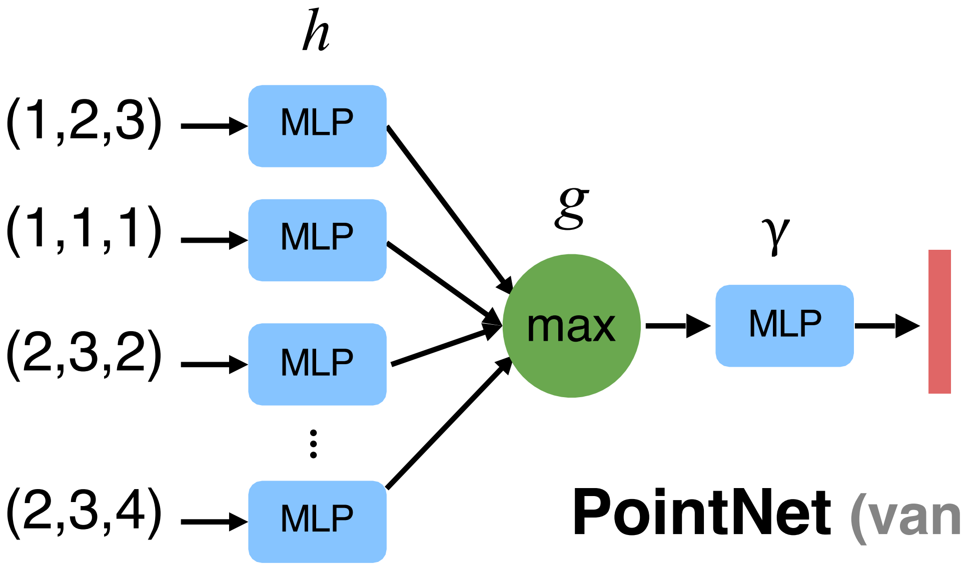
Permutation Invariance: Symmetric Function

What symmetric functions can be constructed by PointNet?



Basic PointNet Architecture

Empirically, we use **multi-layer perceptron (MLP)** and **max pooling**:



PointNet **(vanilla)**

Challenges

Unordered point set as input

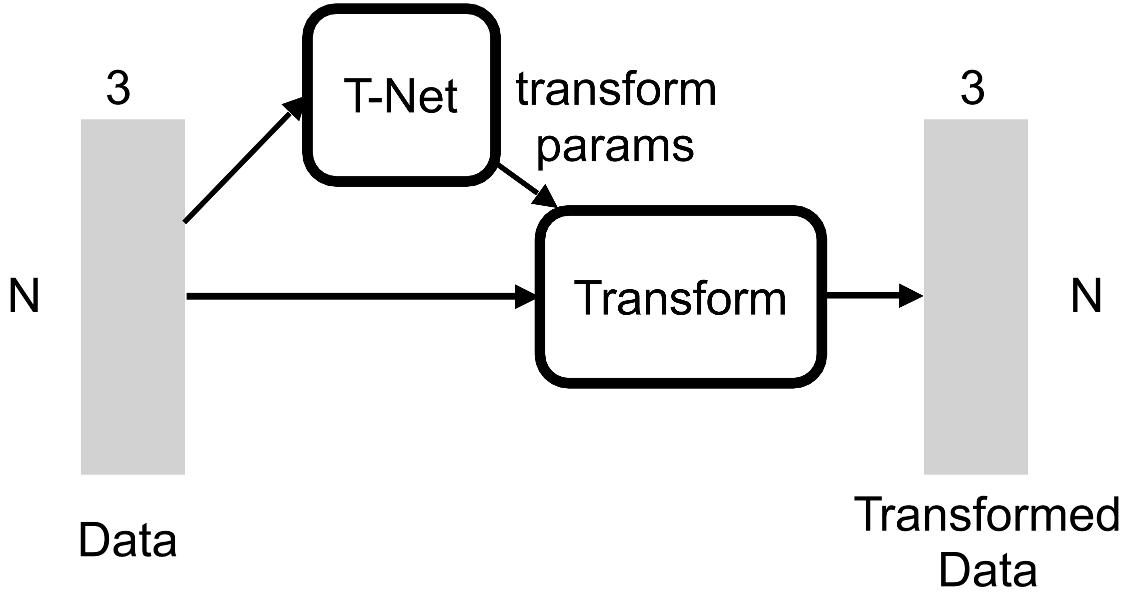
Model needs to be invariant to N! permutations.

Invariance under geometric transformations

Point cloud rotations should not alter classification results.

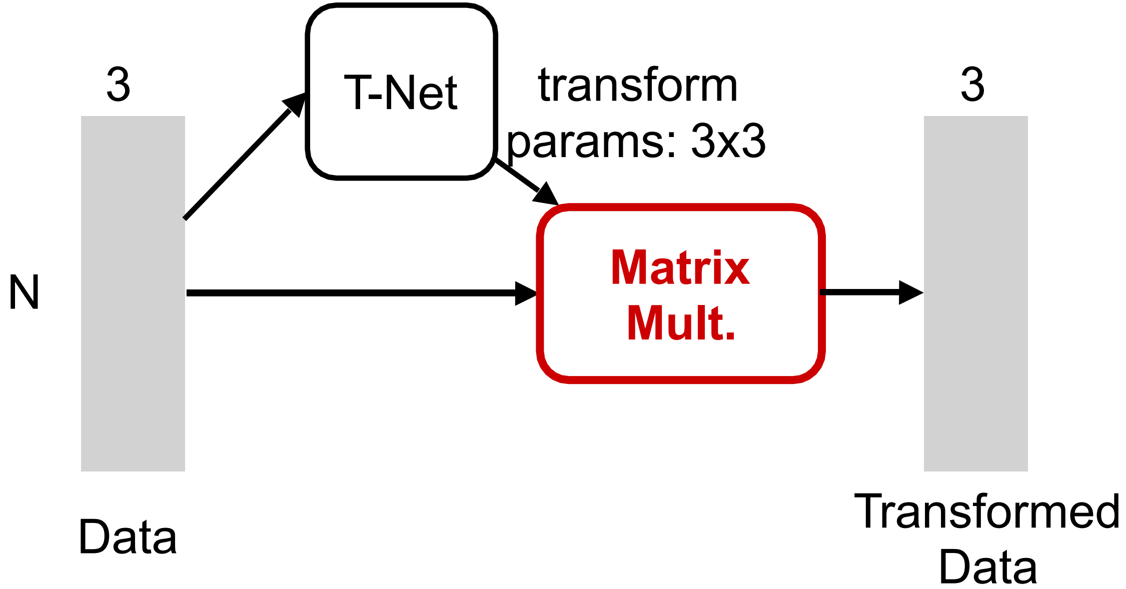
Input Alignment by Transformer Network

Idea: Data dependent transformation for automatic alignment

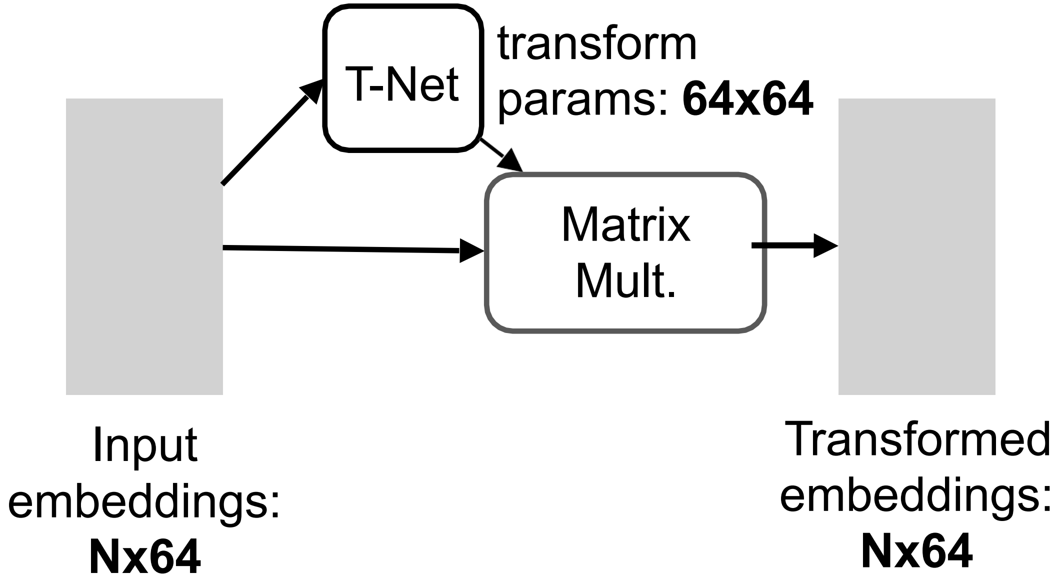


Input Alignment by Transformer Network

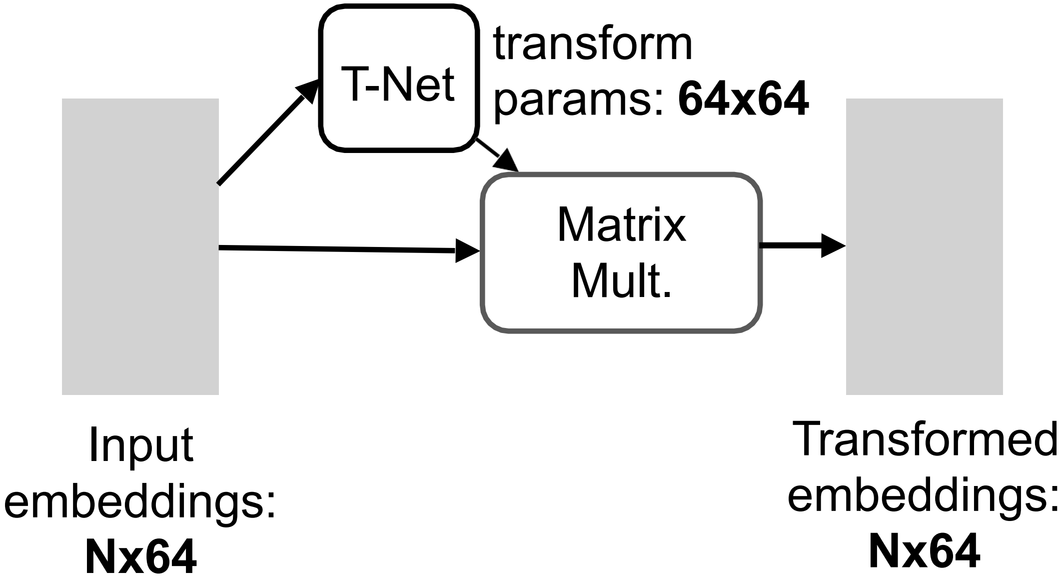
The transformation is just matrix multiplication!



Embedding Space Alignment



Embedding Space Alignment



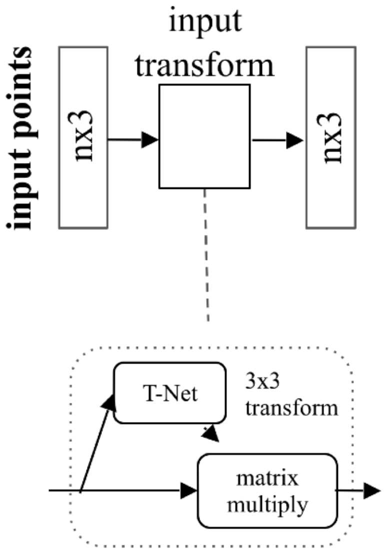
Regularization:

Transform matrix A 64x64 close to orthogonal:

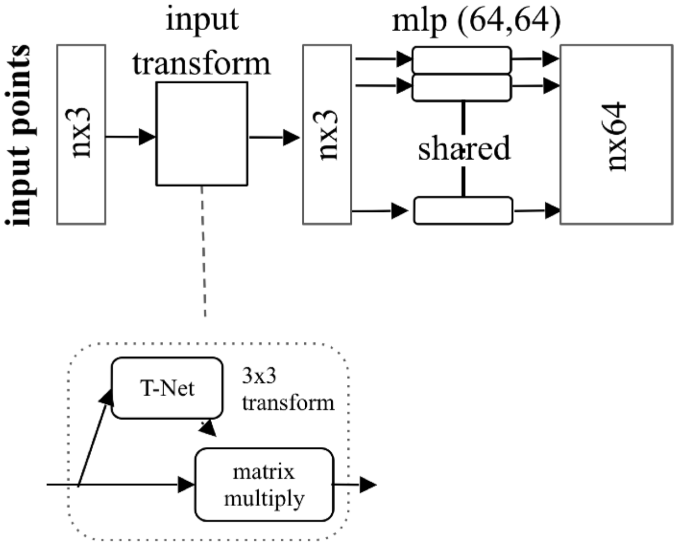
$$L_{reg} = \|I - AA^T\|_F^2$$

PointNet Classification Network

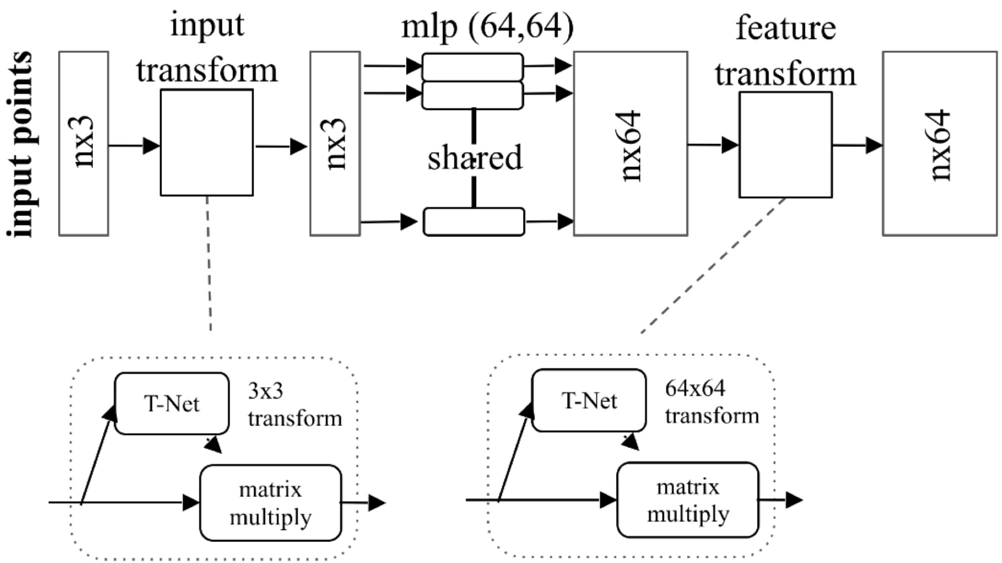
PointNet Classification Network



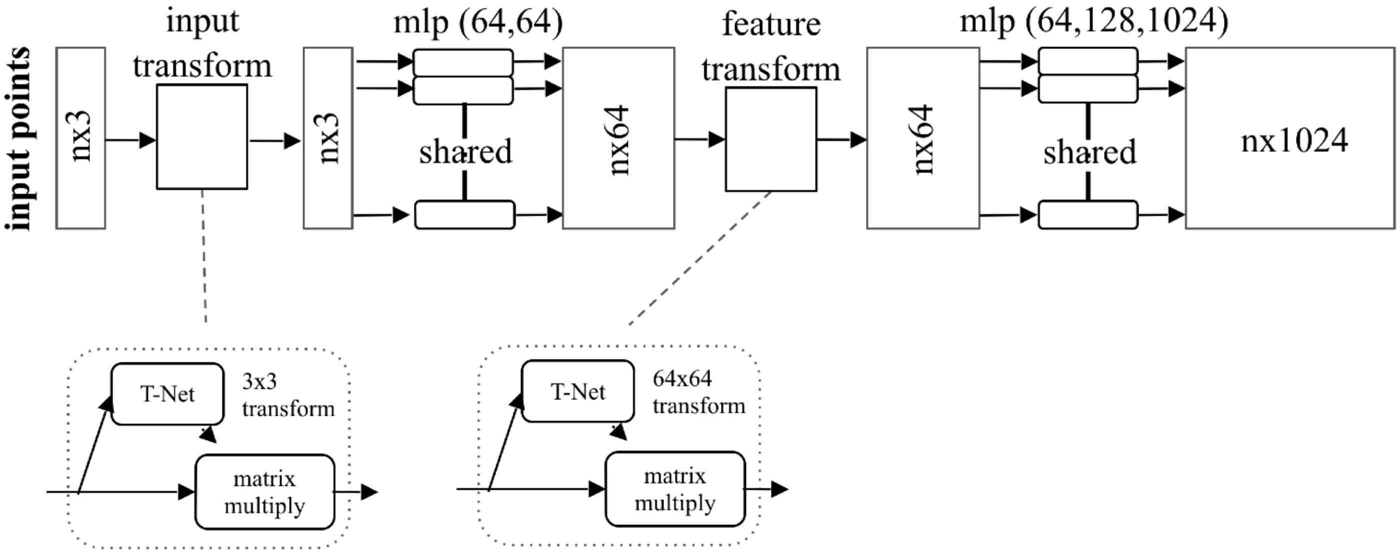
PointNet Classification Network



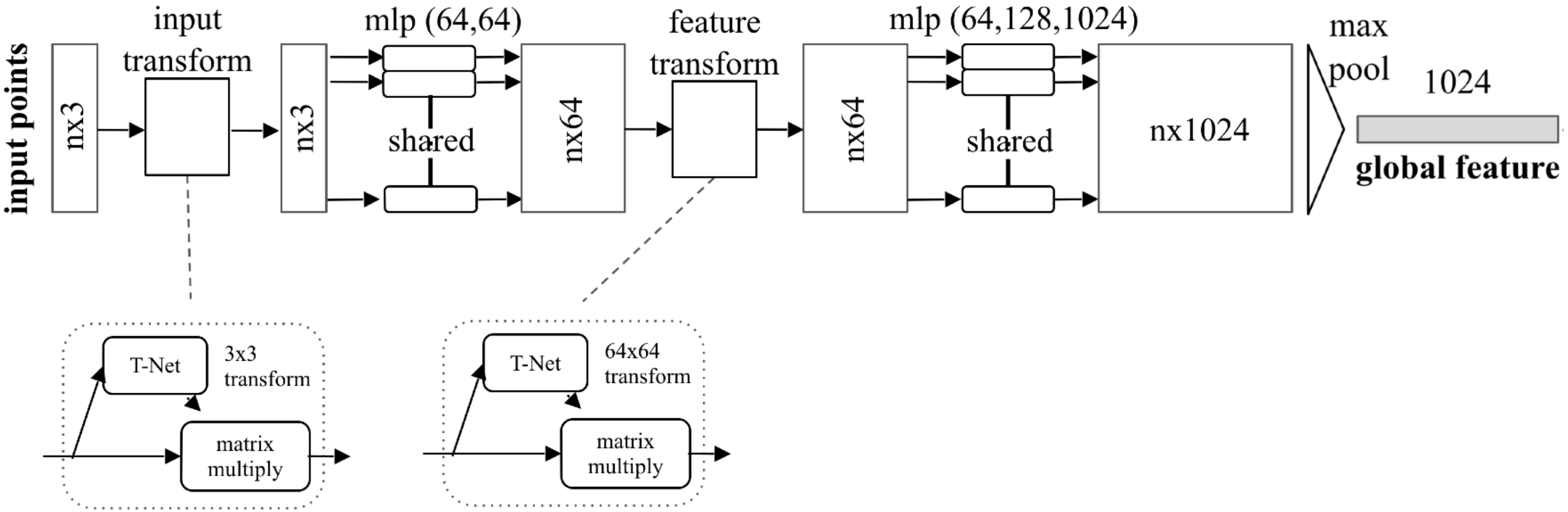
PointNet Classification Network



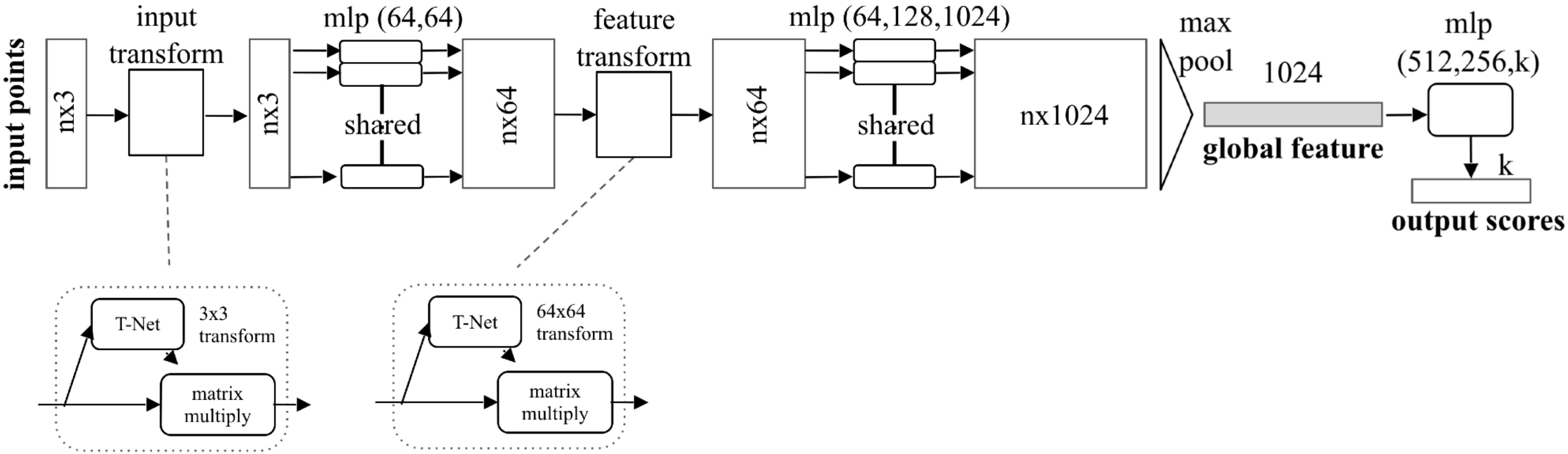
PointNet Classification Network



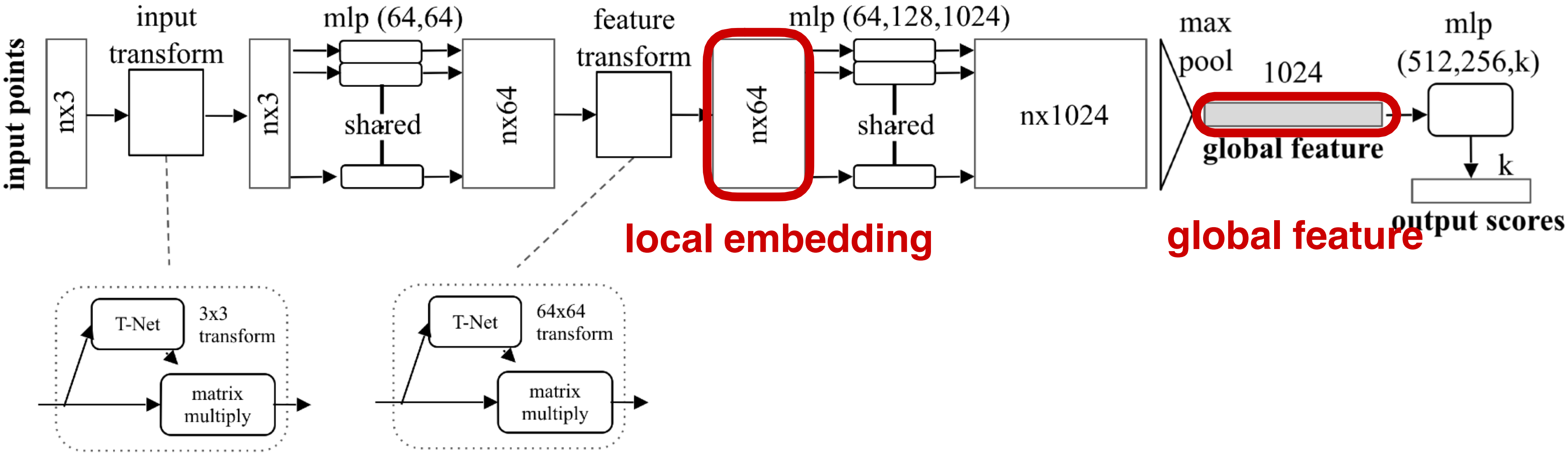
PointNet Classification Network



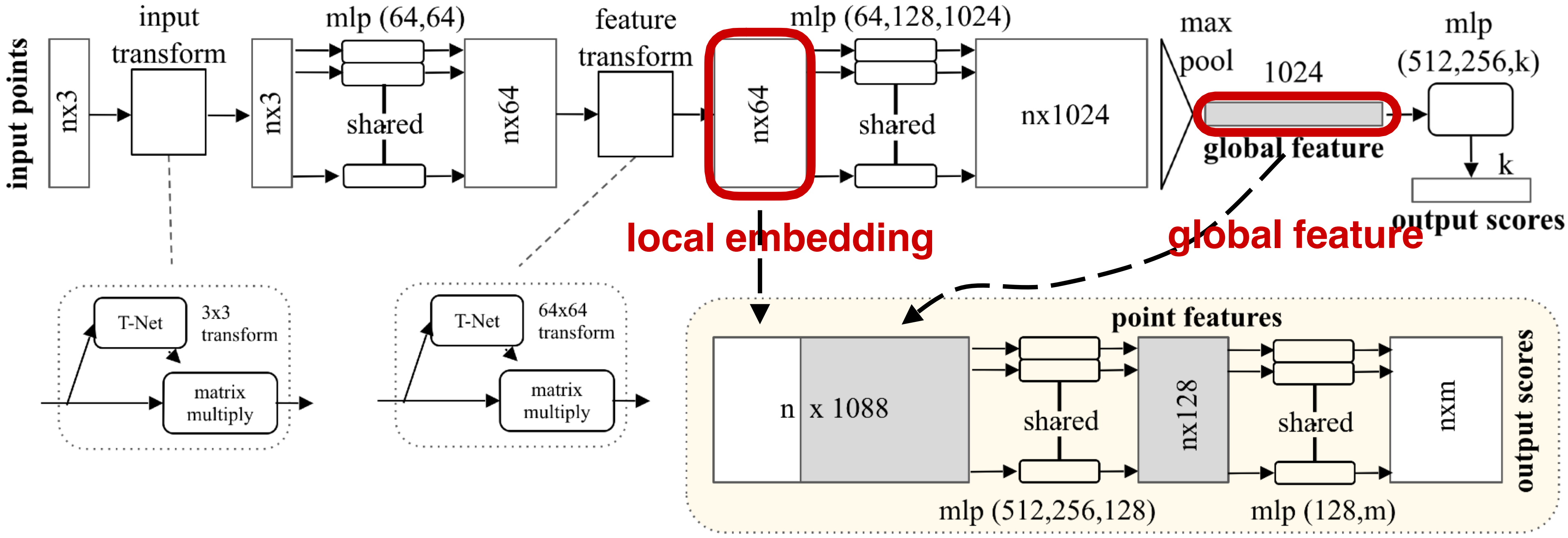
PointNet Classification Network



Extension to PointNet Segmentation Network



Extension to PointNet Segmentation Network



Results

Dataset

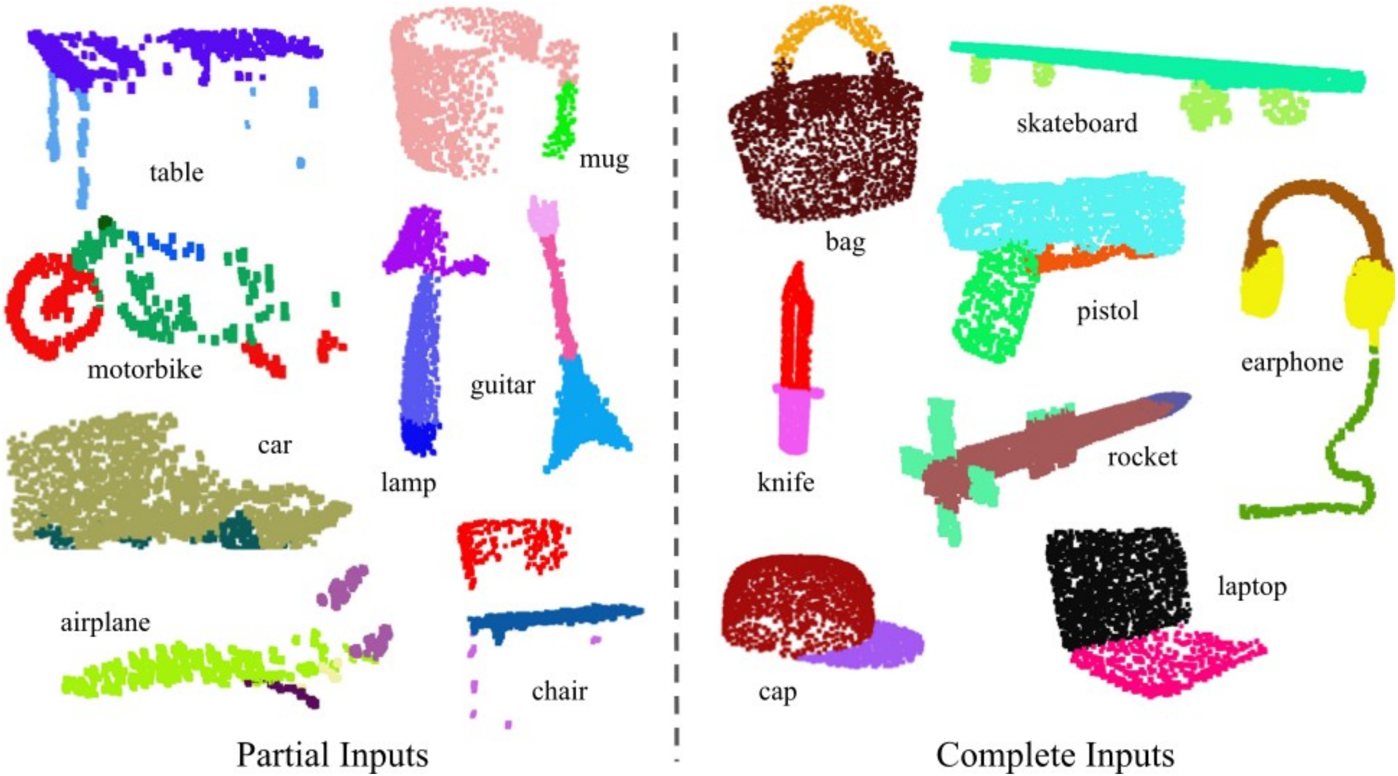
Princeton ModelNet Dataset

A large-scale CAD dataset of more than 150,000 models in 660 categories.

Object Categories

Examples of Chairs

Results on Object Part Segmentation



Results on Object Part Segmentation

	mean	aero	bag	cap	car	chair	ear phone	guitar	knife	lamp	laptop	motor	mug	pistol	rocket	skate board	table
# shapes		2690	76	55	898	3758	69	787	392	1547	451	202	184	283	66	152	5271
Wu [28]	-	63.2	-	-	-	73.5	-	-	-	74.4	-	-	-	-	-	-	74.8
Yi [30]	81.4	81.0	78.4	77.7	**75.7**	87.6	61.9	**92.0**	85.4	**82.5**	**95.7**	**70.6**	91.9	**85.9**	53.1	69.8	75.3
3DCNN	79.4	75.1	72.8	73.3	70.0	87.2	63.5	88.4	79.6	74.4	93.9	58.7	91.8	76.4	51.2	65.3	77.1
Ours	**83.7**	**83.4**	**78.7**	**82.5**	74.9	**89.6**	**73.0**	91.5	**85.9**	80.8	95.3	65.2	**93.0**	81.2	**57.9**	**72.8**	**80.6**

dataset: ShapeNetPart; metric: mean IoU (%)

Results on Semantic Scene Parsing

dataset: Stanford 2D-3D-S (Matterport scans)

Robustness to Data Corruption

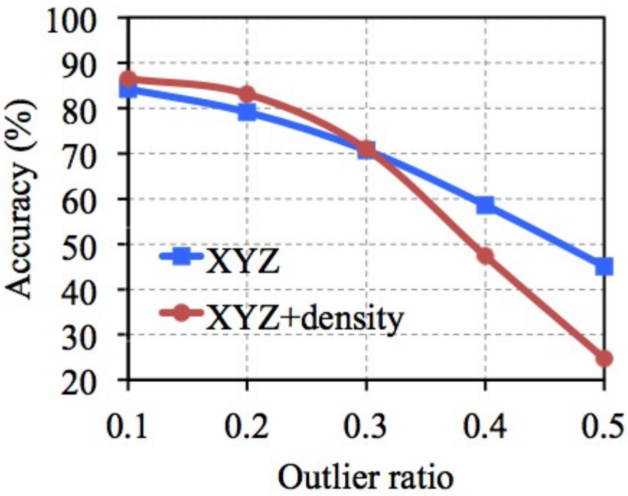
dataset: ModelNet40; metric: 40-class classification accuracy (%)

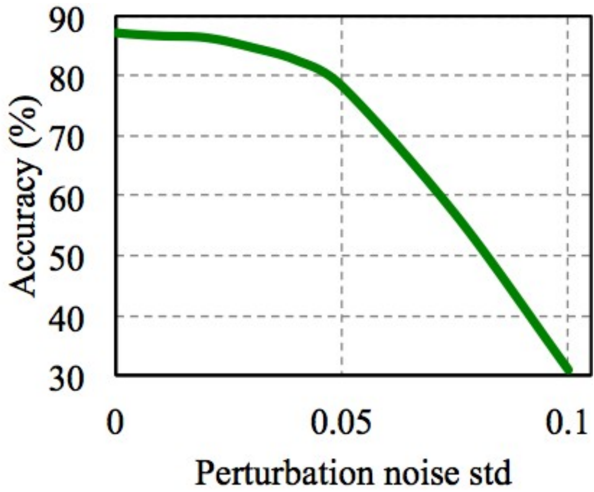
Robustness to Data Corruption

Less than 2% accuracy drop with 50% missing data

dataset: ModelNet40; metric: 40-class classification accuracy (%)

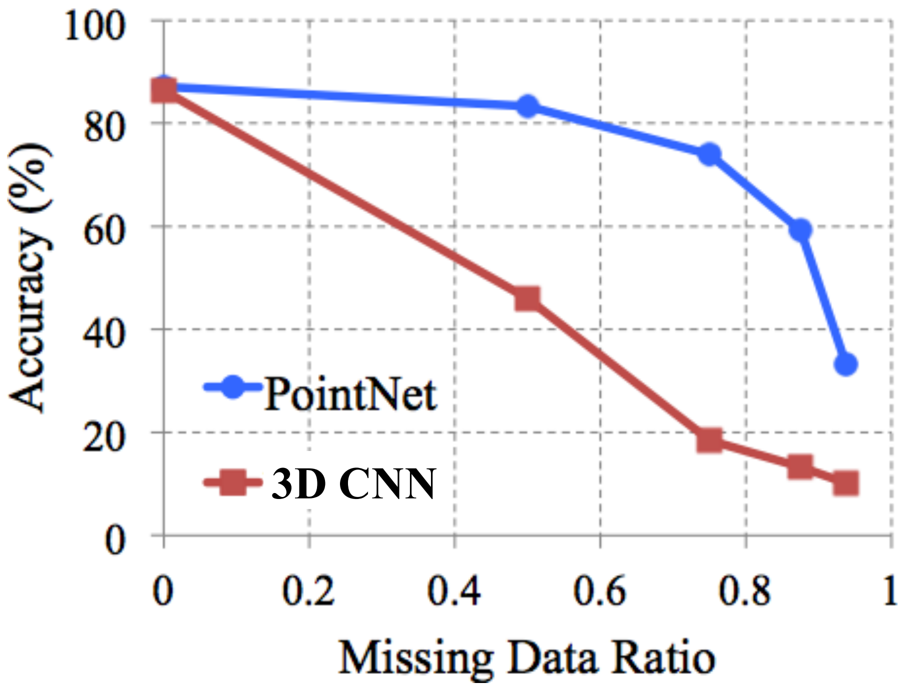
Robustness to Data Corruption





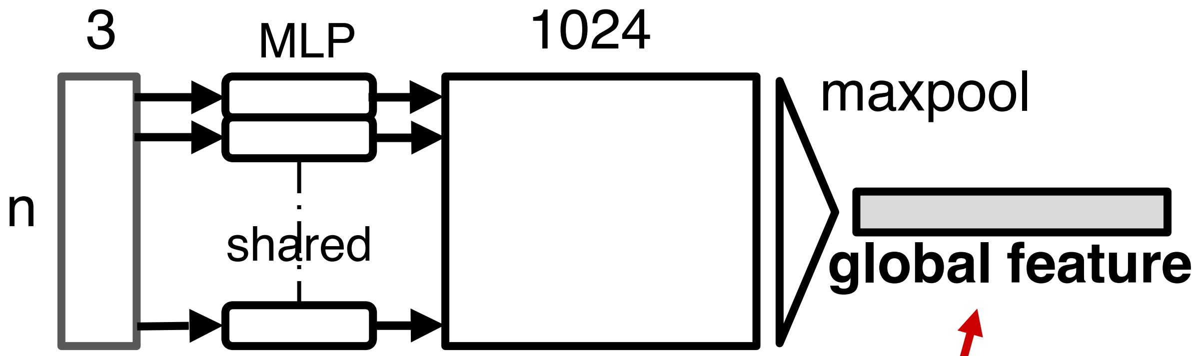
dataset: ModelNet40; metric: 40-class classification accuracy (%)

Robustness to Data Corruption



Why is PointNet so robust to missing data?

Visualizing Global Point Cloud Features



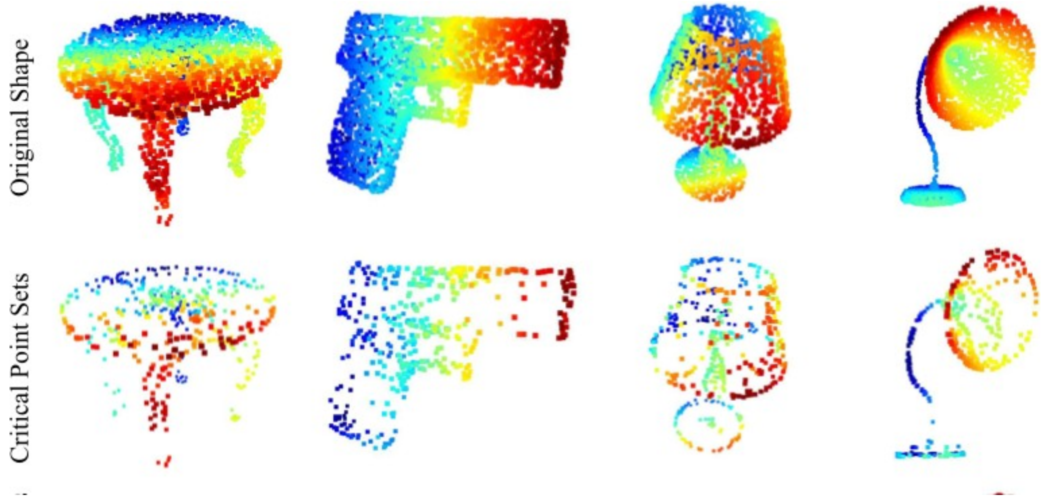
Which input points are contributing to the global feature?

(critical points)

Visualizing Global Point Cloud Features

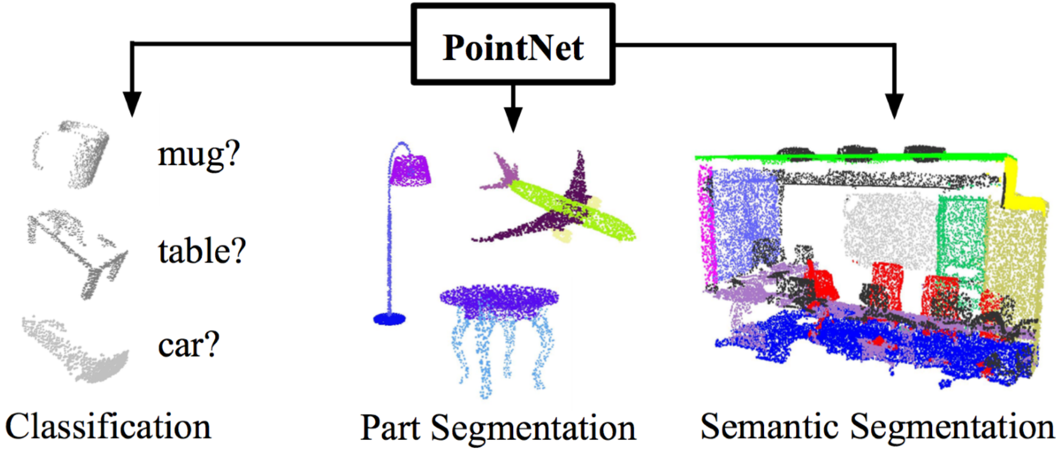
Original Shape:

Critical Point Sets:



Conclusion

- PointNet is a novel deep neural network that directly consumes point cloud.
- A unified approach to various 3D recognition tasks.
- Rich theoretical analysis and experimental results.



Code & Data Available!
http://stanford.edu/~rqi/pointnet

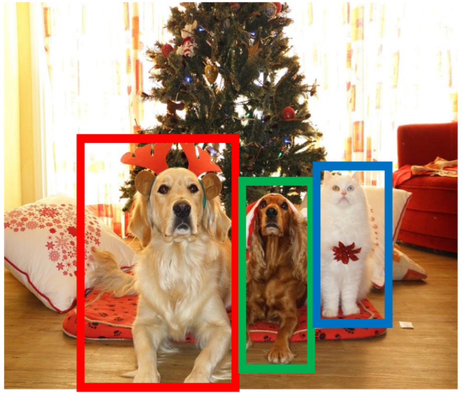
More Computer Vision Tasks

2D Semantic Segmentation

GRASS, CAT, TREE, SKY

Object categories + 2D segments

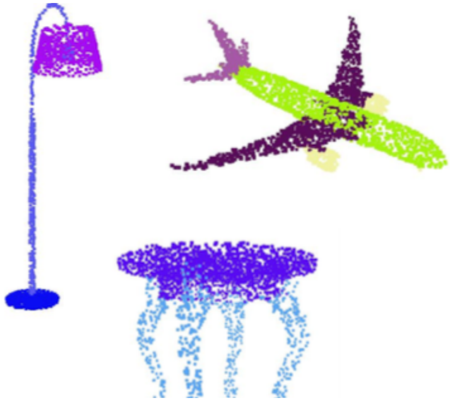
2D Object Detection



DOG, DOG, CAT

Object categories + 2D bounding boxes

3D Classificaion & Segmentation



Object categories + 3D segments

Thanks!