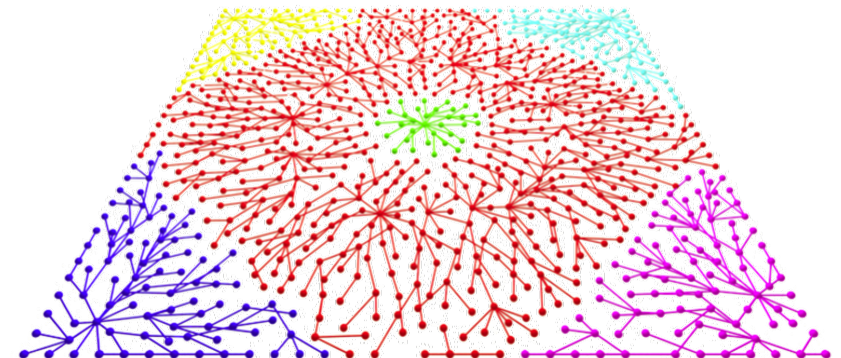


CS233, CME251: Geometric and Topological Data Analysis

Leonidas Guibas
Computer Science Department
Stanford University



Lecture 13
20 May 2020



Last Time: Shape Features,
Alignments, and
Correspondences

Fundamental Registration Problem



Given two shapes with partially overlapping geometry, find an alignment between them

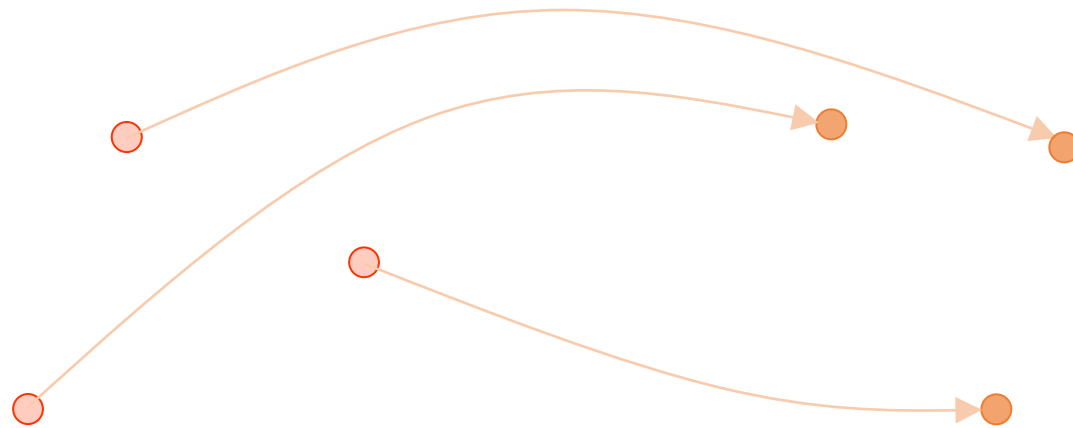
Optimal Rigid Alignment for Points

Problem Formulation:

1. Given two sets points: $\{x_i\}, \{y_i\}, i = 1..n$ in \mathbb{R}^3 Find the rigid transform:

\mathbf{R}, t that minimizes:

$$\sum_{i=1}^N \|\mathbf{R}x_i + t - y_i\|_2^2$$



Optimal Transformation Summary

Problem Formulation:

1. Given two sets points: $\{x_i\}, \{y_i\}, i = 1..n$ in \mathbb{R}^3 . Find the rigid transform:

\mathbf{R}, t that minimizes:
$$\sum_{i=1}^N \|\mathbf{R}x_i + t - y_i\|_2^2$$

2. Closed form solution:

1. Construct: $C = \sum_{i=1}^N (y_i - \mu^Y)(x_i - \mu^X)^T$, where $\mu^X = \frac{1}{N} \sum_i x_i$,

2. Compute the SVD of C: $C = U\Sigma V^T$ $\mu^Y = \frac{1}{N} \sum_i y_i$

1. If $\det(UV^T) = 1, R_{\text{opt}} = UV^T$

2. Else $R_{\text{opt}} = U\tilde{\Sigma}V^T, \tilde{\Sigma} = \text{diag}(1, 1, \dots, -1)$

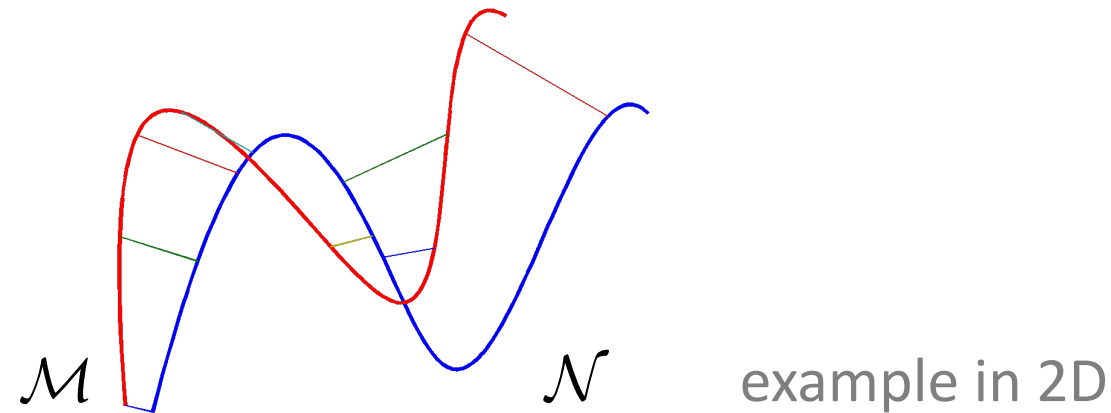
3. Set $t_{\text{opt}} = \mu^Y - R_{\text{opt}}\mu^X$

Note that C is a 3x3 matrix. SVD is very fast.

Arun et al., Least-Squares Fitting of
Two 3-D Point Sets

Iterative Closest Point (ICP)

- Classical approach: iterate between finding correspondences and finding the transformation:



Given a pair of shapes, \mathcal{M} and \mathcal{N} , iterate:

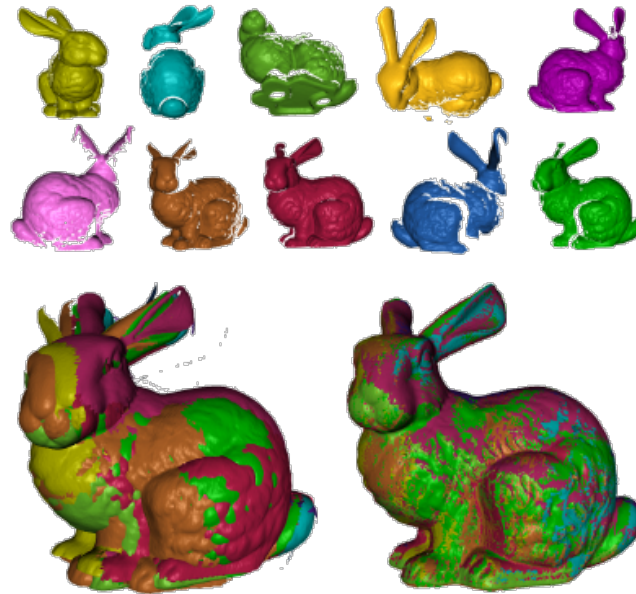
1. For each $x_i \in \mathcal{M}$ find **nearest** neighbor $y_i \in \mathcal{N}$
2. Find optimal transformation \mathbf{R}, t minimizing:

$$\arg \min_{R, t} \sum_i \|Rx_i + t - y_i\|_2^2$$

Classic problem,
solvable by SVD

Global Matching

Given shapes in *arbitrary* positions, find their alignment:

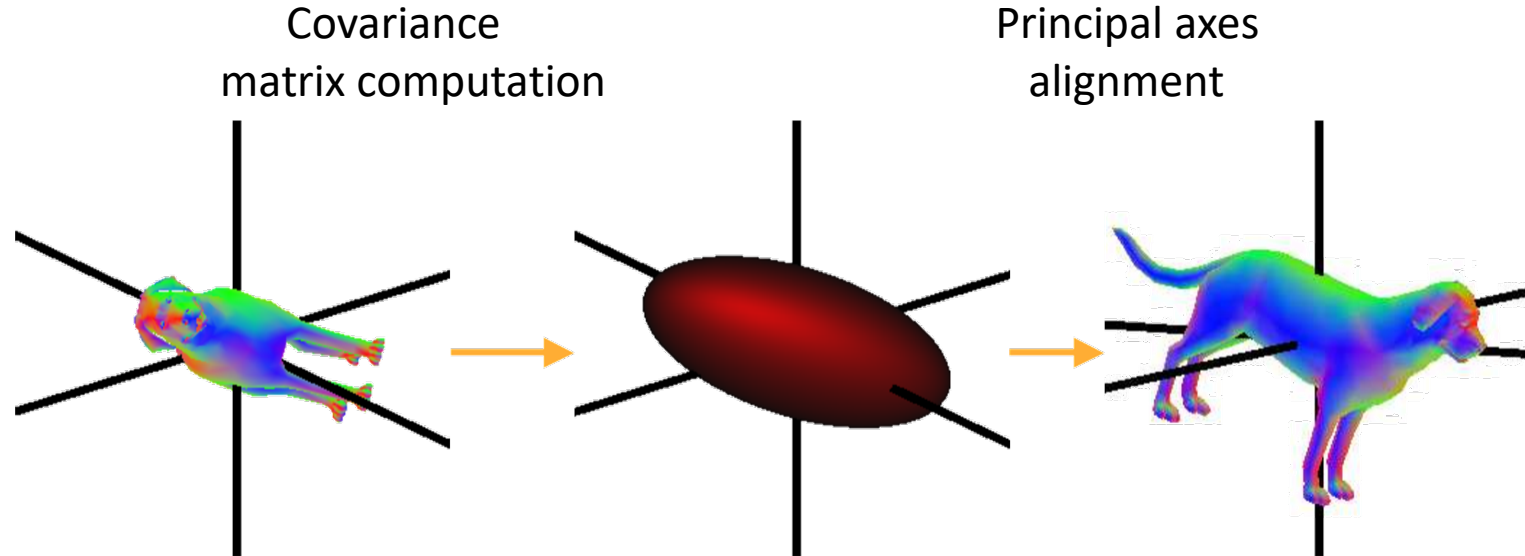


Robust Global Registration
Gelfand et al. SGP 2005

Can be approximate, since will refine later using e.g. ICP

PCA-Based Alignment

- ◆ Use PCA to place models into canonical coordinate frames
- ◆ Then align those frames

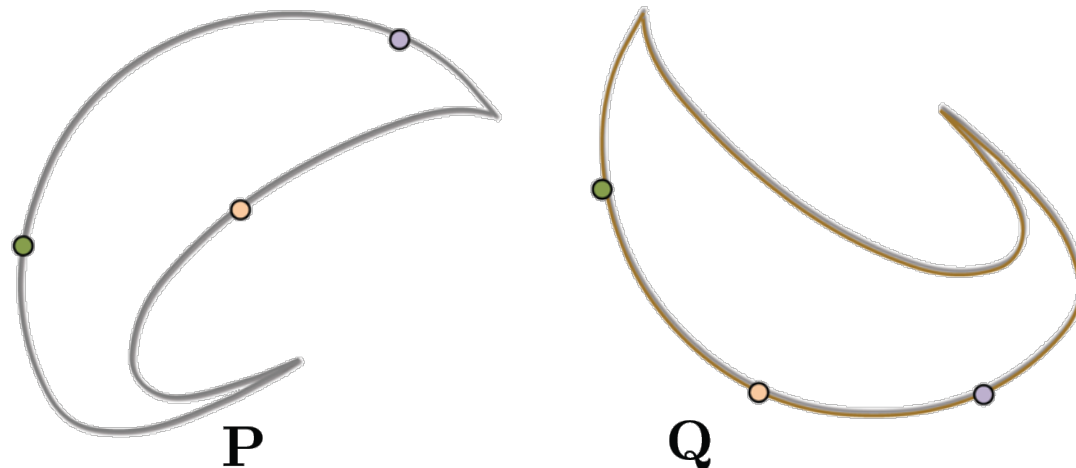


Random Sampling (RANSAC)

ICP only needs 3 point pairs! – Rigid motion space is 6-dimensional.

Robust and Simple approach. Iterate between:

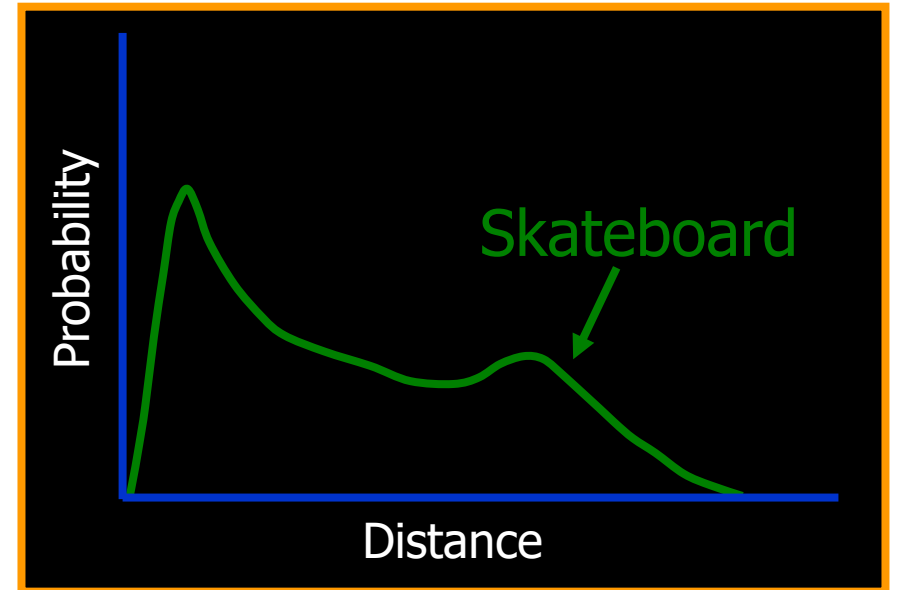
1. Pick a random pair of 3 points on model & scan
2. Estimate alignment, and check for error.



Guess and
verify

D2 Shape Distributions

- Properties
 - Concise to store?
 - Quick to compute?
 - Invariant to transforms?
 - Efficient to match?
 - Insensitive to noise?
 - Insensitive to topology?
 - Robust to degeneracies?
 - Invariant to deformations?
 - Discriminating?

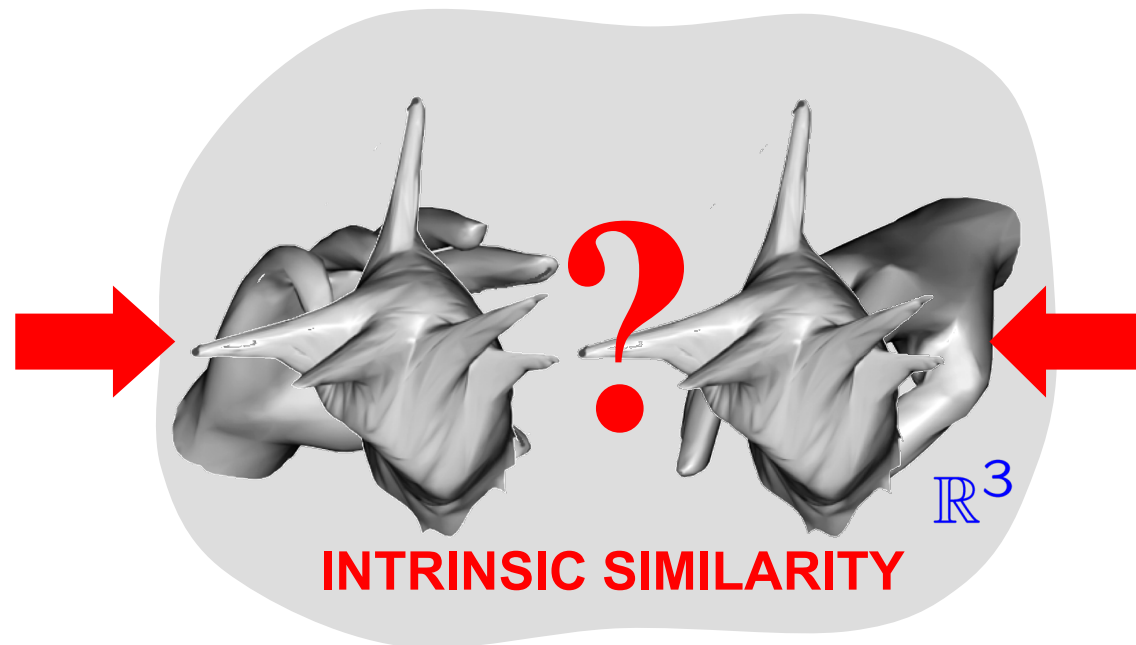


512 bytes (64 values)

0.5 seconds (10^6 samples)

Global Point Signature (GPS)

$$GPS(p) = \left(\frac{1}{\sqrt{\lambda_1}} \phi_1(p), \frac{1}{\sqrt{\lambda_2}} \phi_2(p), \frac{1}{\sqrt{\lambda_3}} \phi_3(p), \dots \right)$$



but there is a sign ambiguity

The Heat Kernel Signature

- Let $k_t(x, \cdot)$ be the signature of x at scale t
The heat kernel has all the properties we want.
Except easy comparison ...

- We define the **Heat Kernel Signature** (HKS), by restricting to the diagonal:

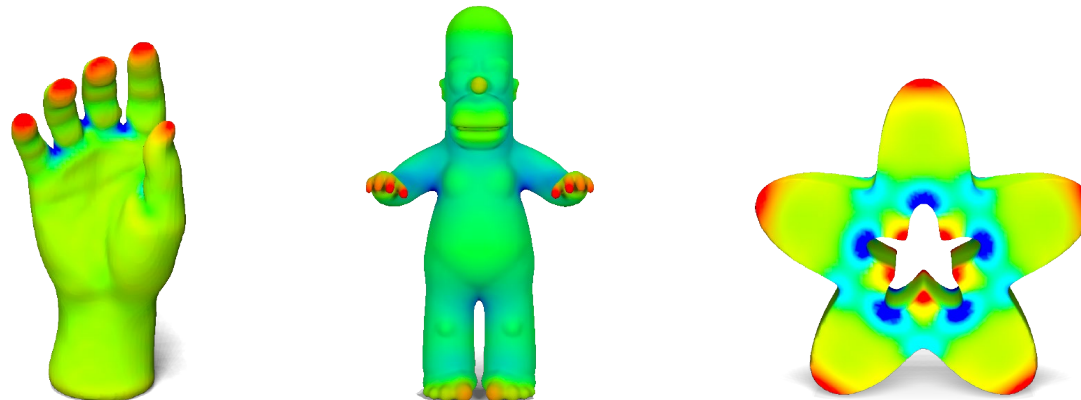
$$\text{HKS}(x) = \{k_t(x, x), t \in \mathbb{R}^+\}$$

- Now HKSs of two points can be easily compared since they are defined on a common domain (time)

Defining a Signature

- Since HKS is a restriction of the heat kernel, it is:
 - Robust
 - Multiscale
- Question: How informative is it?
 - Related to Gaussian curvature for small t :

$$k_t(x, x) = \frac{1}{4\pi t} \sum_{i=0}^{\infty} a_i t^i \quad a_0 = 1, a_1 = \frac{1}{6}K$$



Today: Deep Nets, Multi-View and Volumetric Approaches to 3D

Agenda

- ◆ Today
 - ◆ Deep Learning Intro
 - ◆ 3D Deep Learning
 - ◆ Multi-view CNNs
 - ◆ Volumetric CNNs

Machine Learning

◆ Traditional Programming

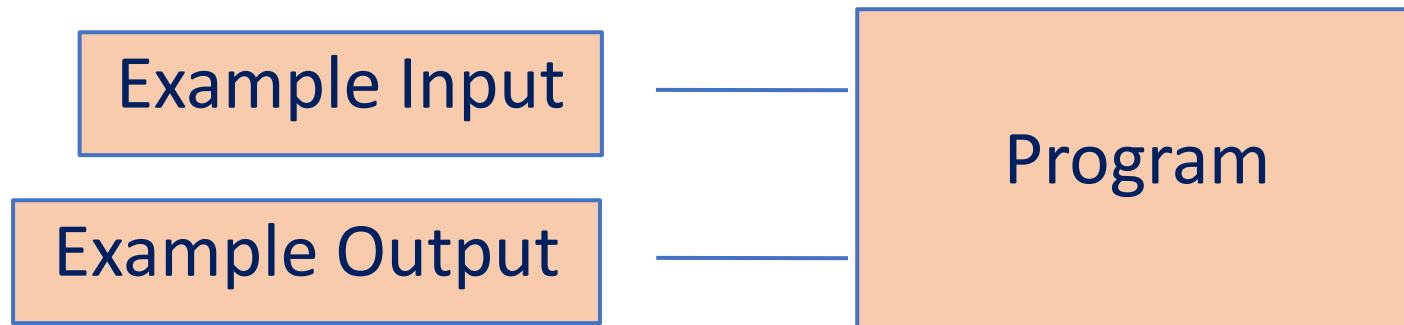


Machine Learning

◆ Traditional Programming



◆ Machine Learning



Machine Learning

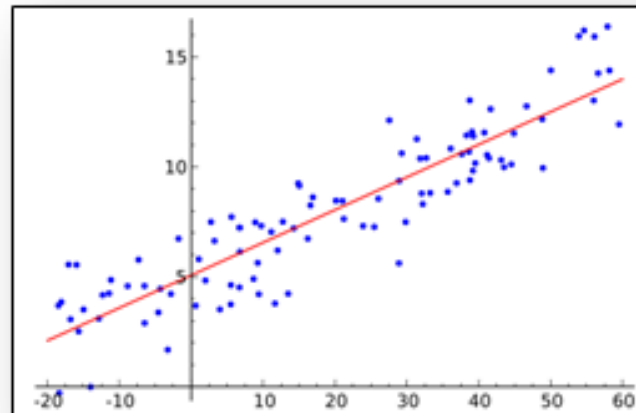
Parametrized by many
learnable parameters

$f(\$

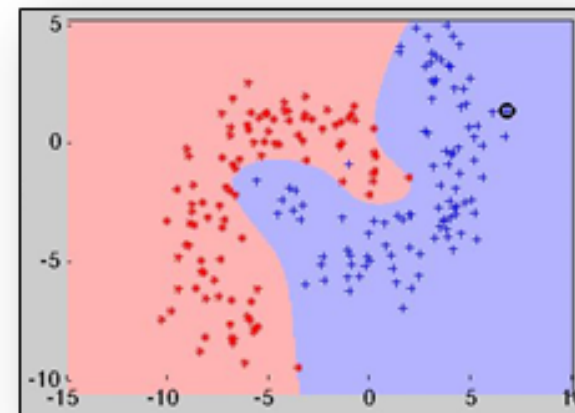


$)=cat$

Model Fitting



Regression

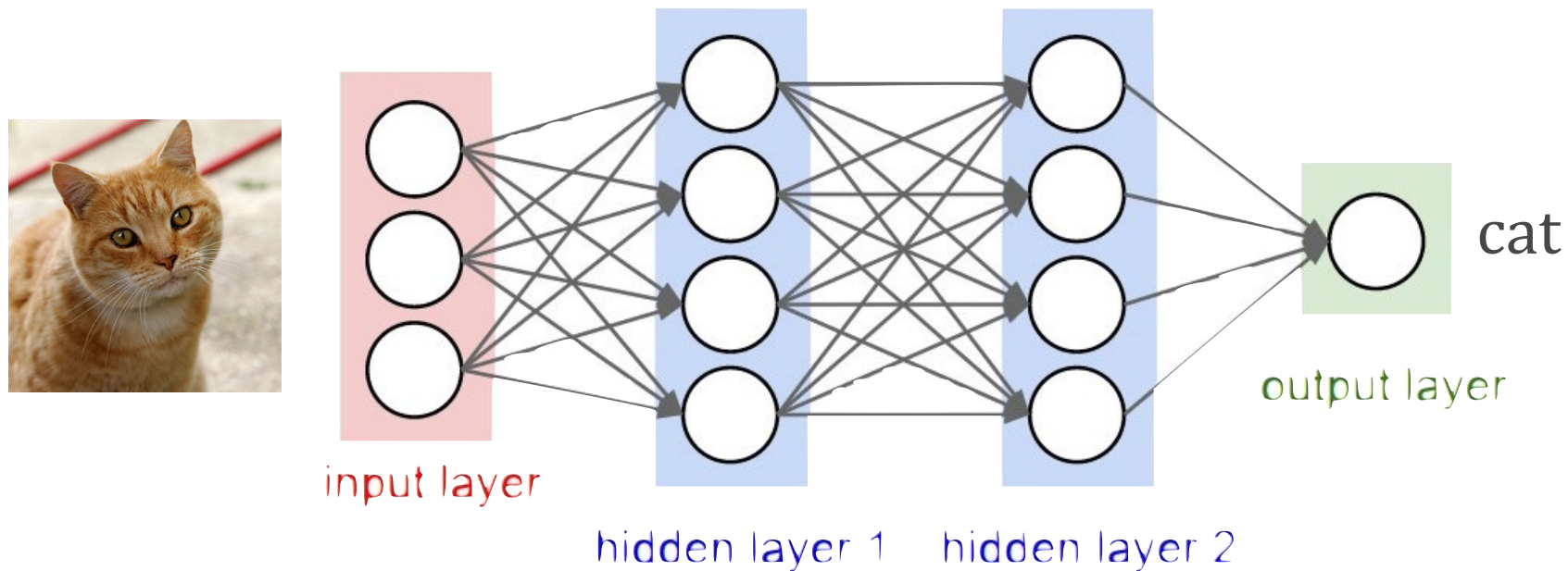


Classification

Deep Learning

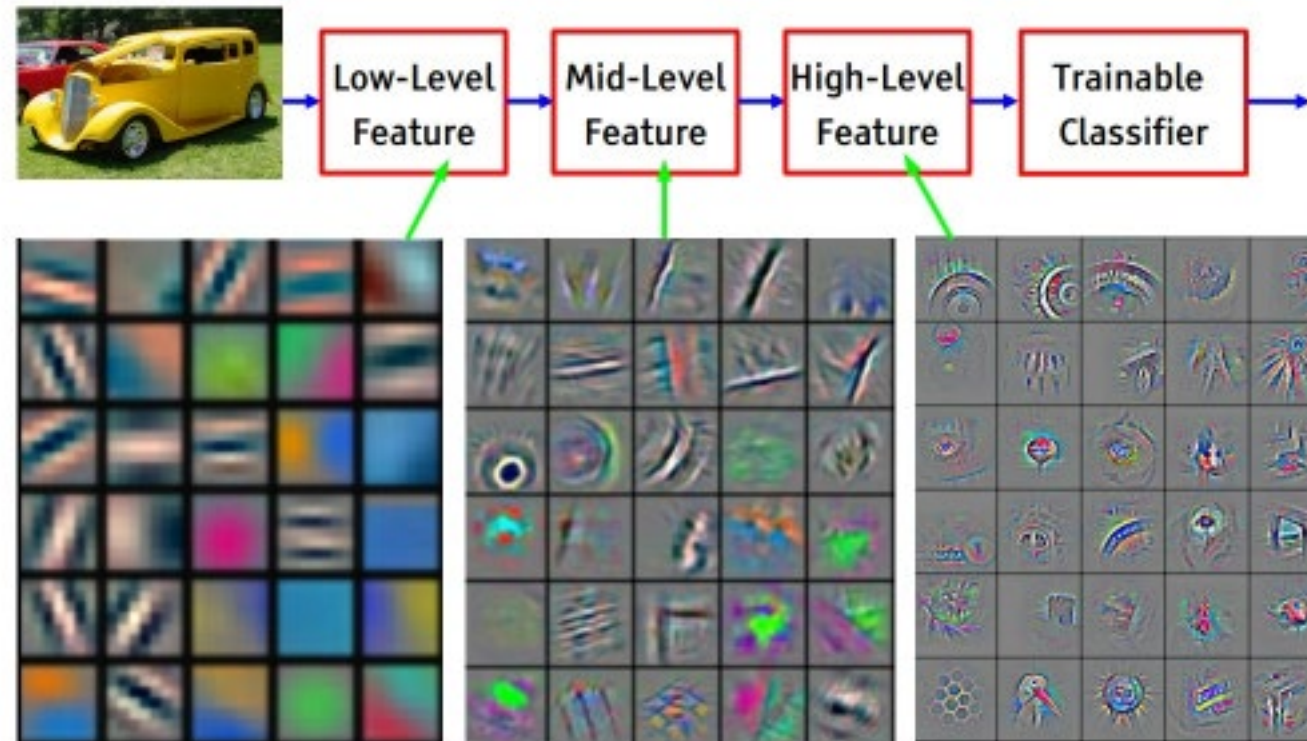
- ◆ Deep learning allows computational models that are composed of **multiple processing layers to learn representations of data** with **multiple levels of abstraction**.

Deep Learning by Y. LeCun et al. Nature 2015



Neural Networks as Feature Extractors

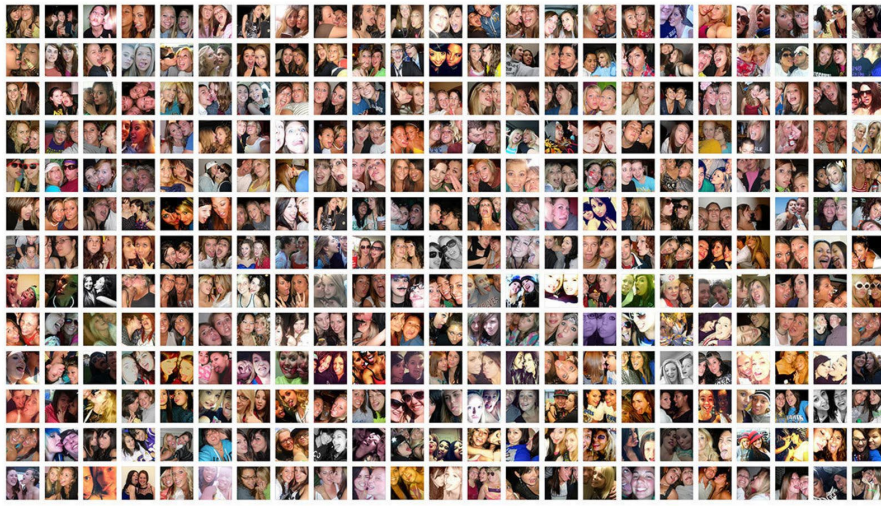
- ◆ Neural networks extract powerful features from data



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Image Credits: Yan LeCun

Deep Learning: Powered By

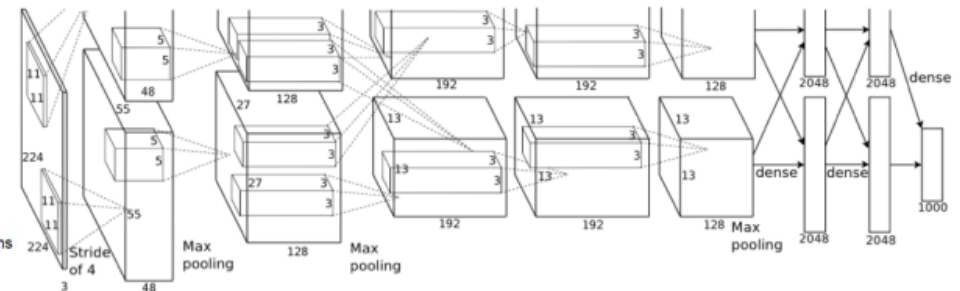
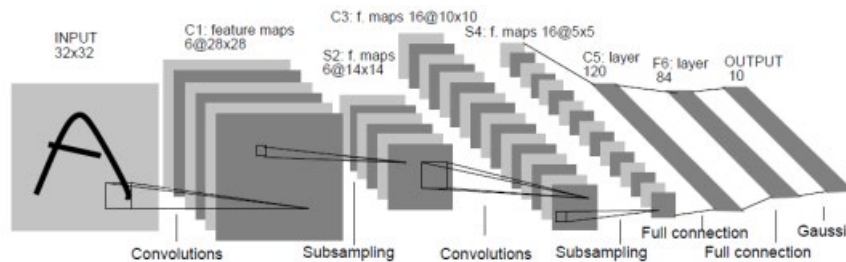


Lots of data



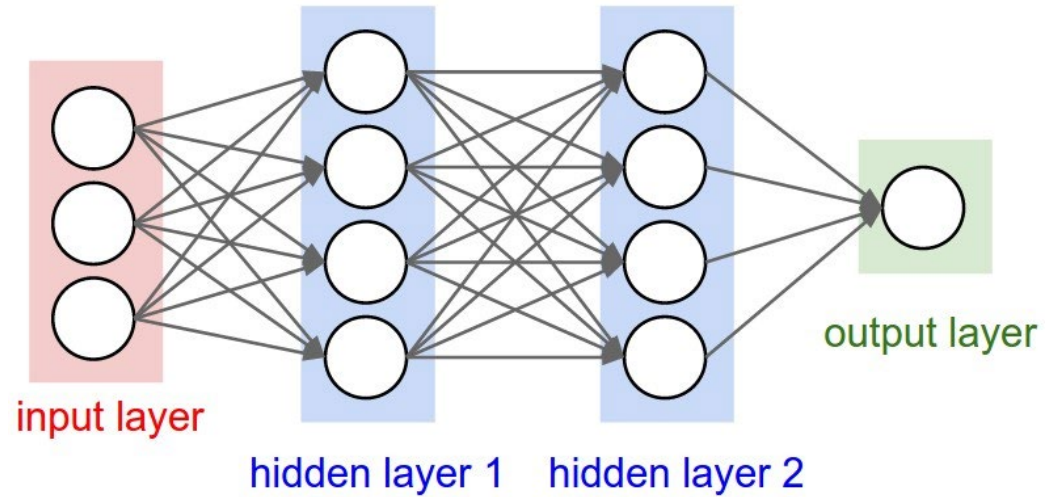
Lots of computing power

+



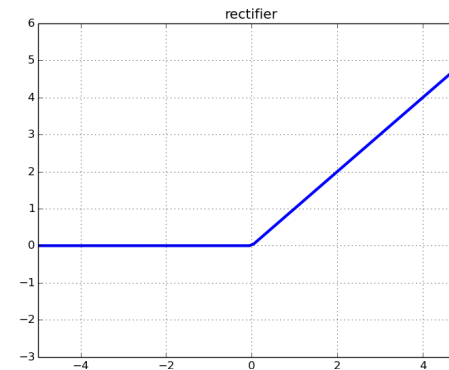
Powerful Neural Network Architectures

Neural Networks



f: non-linear activation function

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$



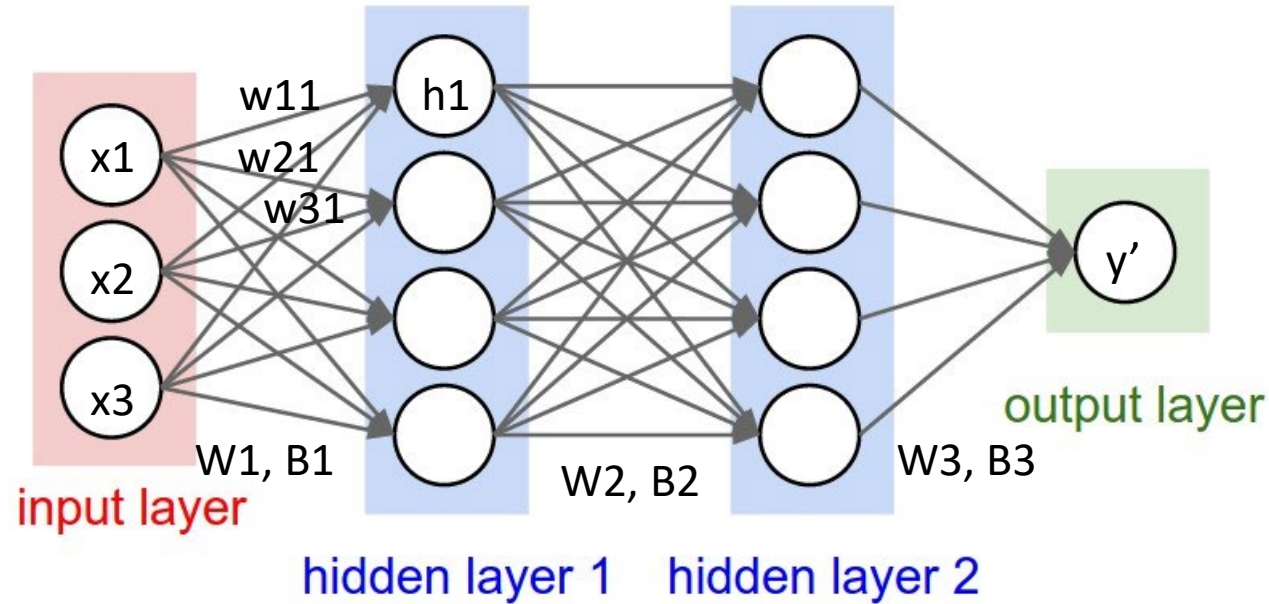
Model: Multi-Layer Perceptron (MLP)

$$y' = W_3 f(W_2 f(W_1 x + b_1) + b_2) + b_3$$

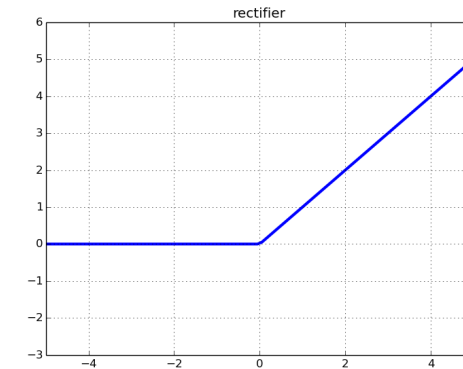
Neural Networks

$$h_1 = f(w_{11} * x_1 + w_{21} * x_2 + w_{31} * x_3 + b_1)$$

f: non-linear activation function



$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$



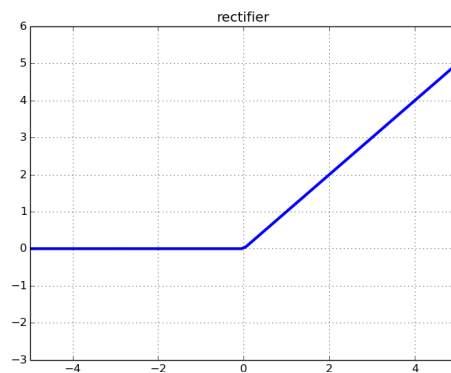
Model: Multi-Layer Perceptron (MLP)

$$y' = W_3 f(W_2 f(W_1 x + b_1) + b_2) + b_3$$

Neural Networks

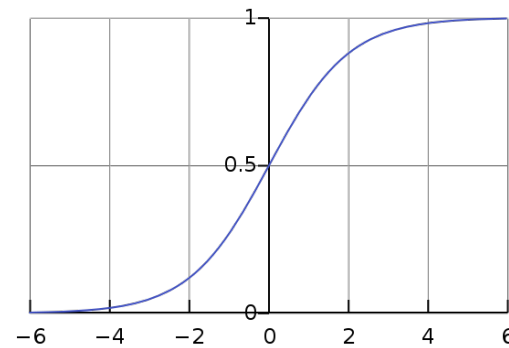
f: non-linear activation function

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

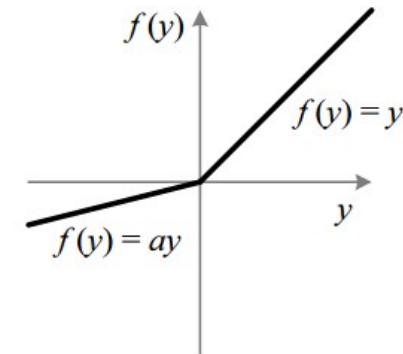


Sigmoid Function

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



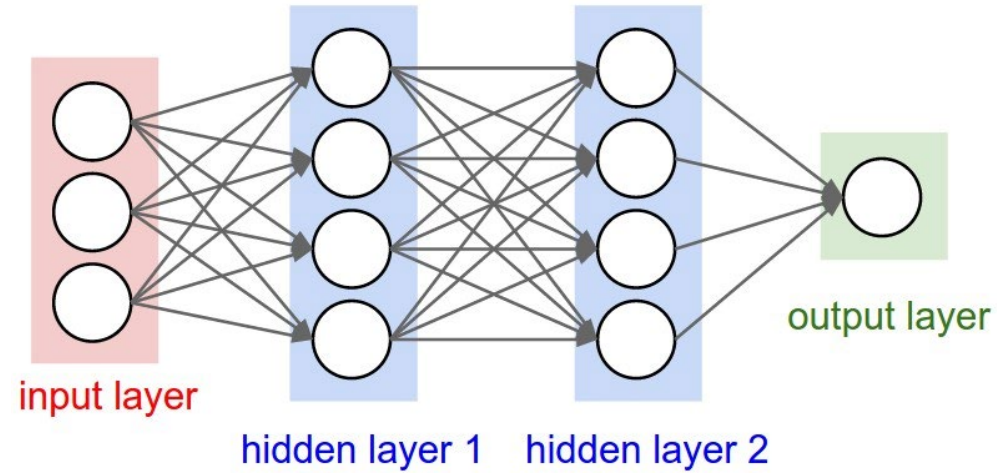
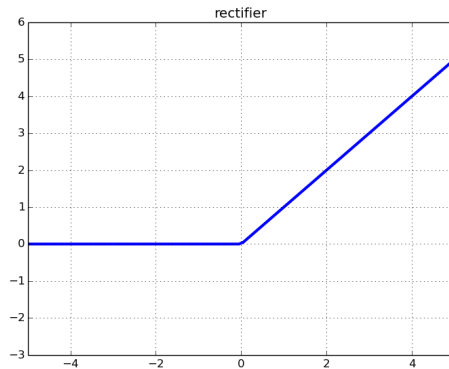
Leaky ReLU



Neural Networks

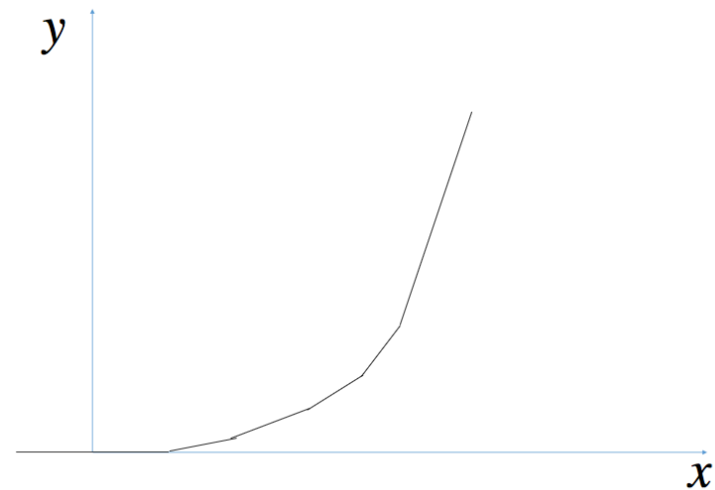
f: non-linear activation function

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$

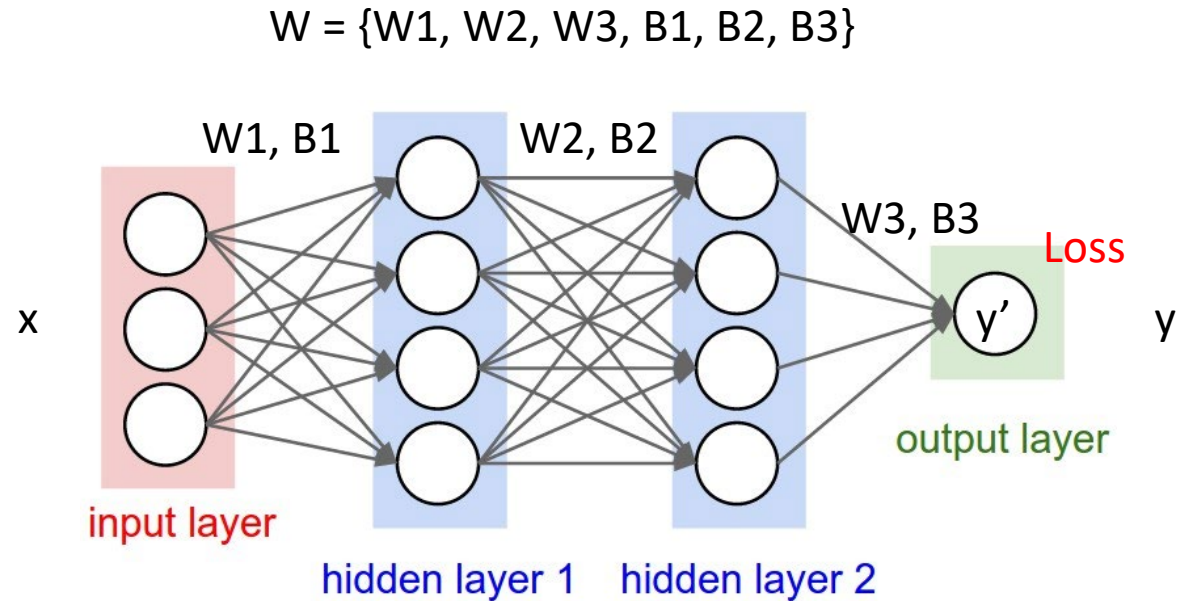


Piece-wise Approximation

$$y = x^2$$

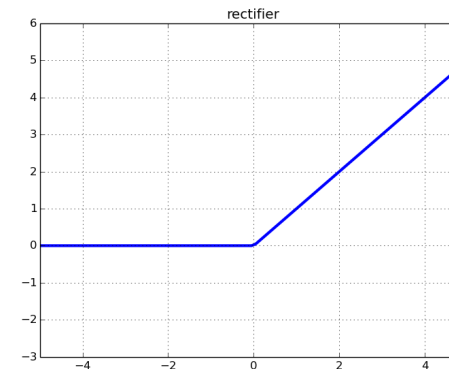


Neural Networks



f: non-linear activation function

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$



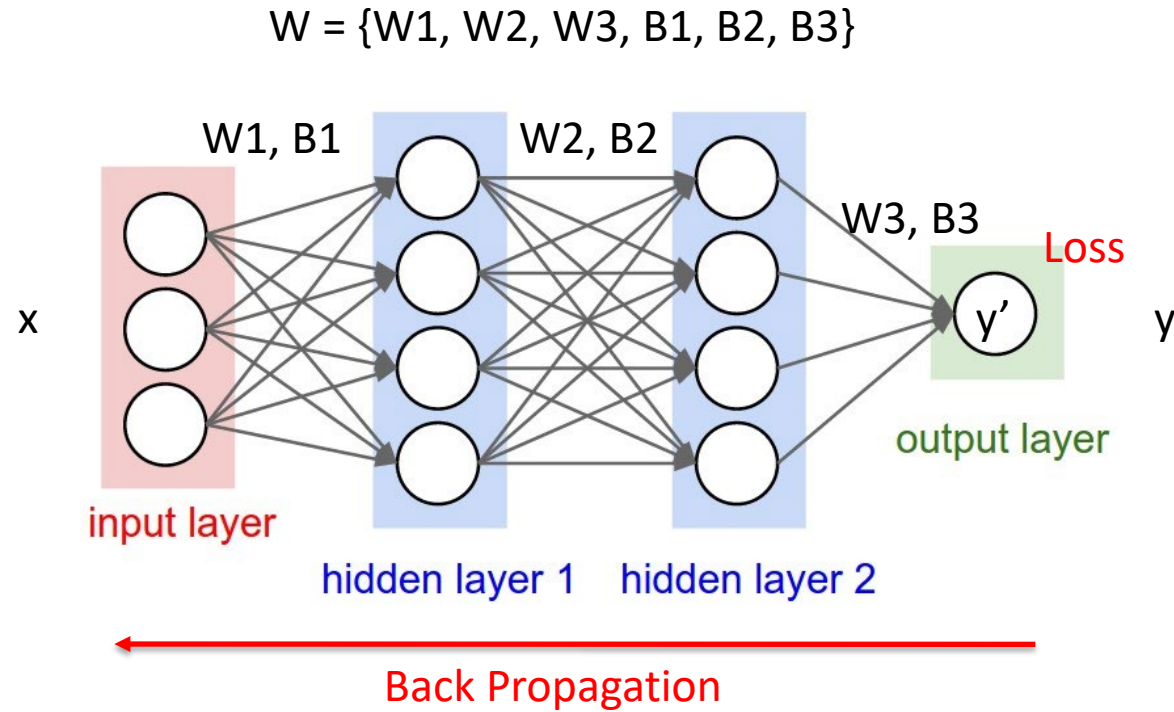
Model: Multi-Layer Perceptron (MLP)

$$y' = W_3 f(W_2 f(W_1 x + b_1) + b_2) + b_3$$

Loss function: L2 loss

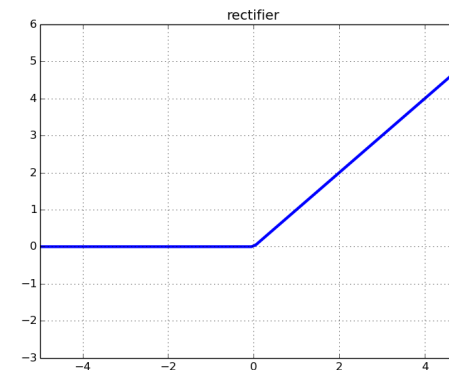
$$l(y, y') = (y - y')^2$$

Neural Networks



f: non-linear activation function

$$\text{ReLU}(x) = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$$



Model: Multi-Layer Perceptron (MLP)

$$y' = W_3 f(W_2 f(W_1 x + b_1) + b_2) + b_3$$

Loss function: L2 loss

$$l(y, y') = (y - y')^2$$

Optimization: Gradient descent

$$W = W - \eta \frac{\partial L}{\partial W}$$

Neural Networks

A three-layer network approximates any continuous function

Let $\varphi(\cdot)$ be a nonconstant, **bounded**, and **monotonically**-increasing **continuous** function. Let I_m denote the m -dimensional **unit hypercube** $[0, 1]^m$. The space of continuous functions on I_m is denoted by $C(I_m)$. Then, given any function $f \in C(I_m)$ and $\varepsilon > 0$, there exists an integer N , real constants $v_i, b_i \in \mathbb{R}$ and real vectors $w_i \in \mathbb{R}^m$, where $i = 1, \dots, N$, such that we may define:

$$F(x) = \sum_{i=1}^N v_i \varphi(w_i^T x + b_i)$$

as an approximate realization of the function f where f is independent of φ ; that is,

$$|F(x) - f(x)| < \varepsilon$$

for all $x \in I_m$. In other words, functions of the form $F(x)$ are **dense** in $C(I_m)$.

Neural Networks

A three-layer network approximates any continuous function

Let $\varphi(\cdot)$ be a nonconstant, **bounded**, and **monotonically-increasing continuous** function. Let I_m denote the m -dimensional **unit hypercube** $[0, 1]^m$. Then, any continuous function f on I_m is in $C(I_m)$. Then, given any function $f \in C(I_m)$, there exists a function $F(x)$ of the form $F(x) = \sum_{i=1}^N v_i \varphi(w_i \cdot x)$, where $i = 1, \dots, N$, and $w_i \in \mathbb{R}^m$,

$$F(x) = \sum_{i=1}^N v_i \varphi(w_i \cdot x)$$

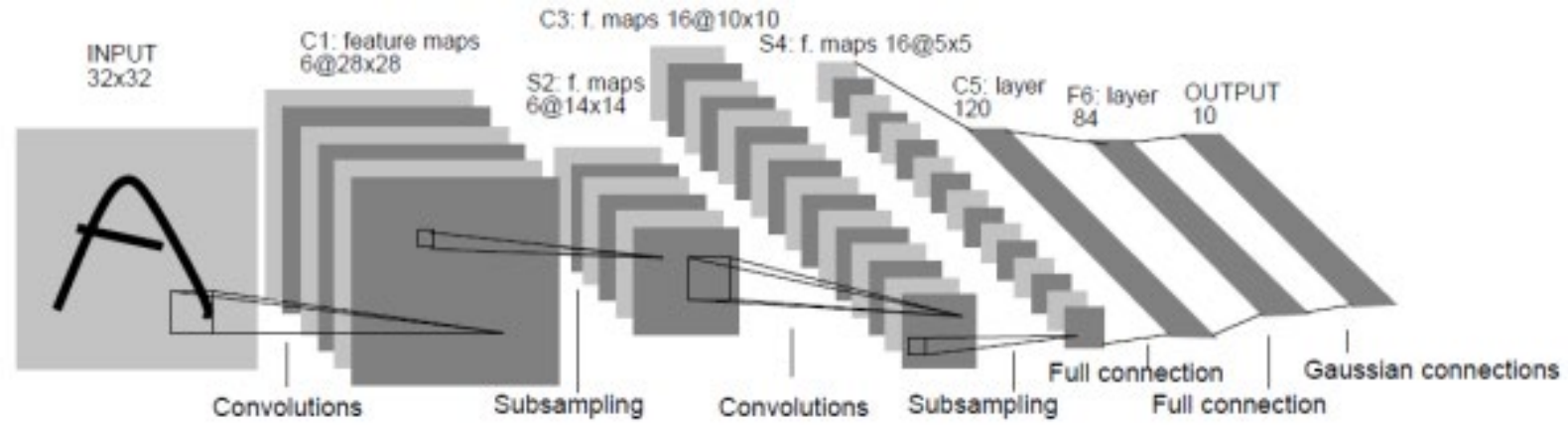
as an approximate

$$|F(x) - f(x)| < \epsilon$$

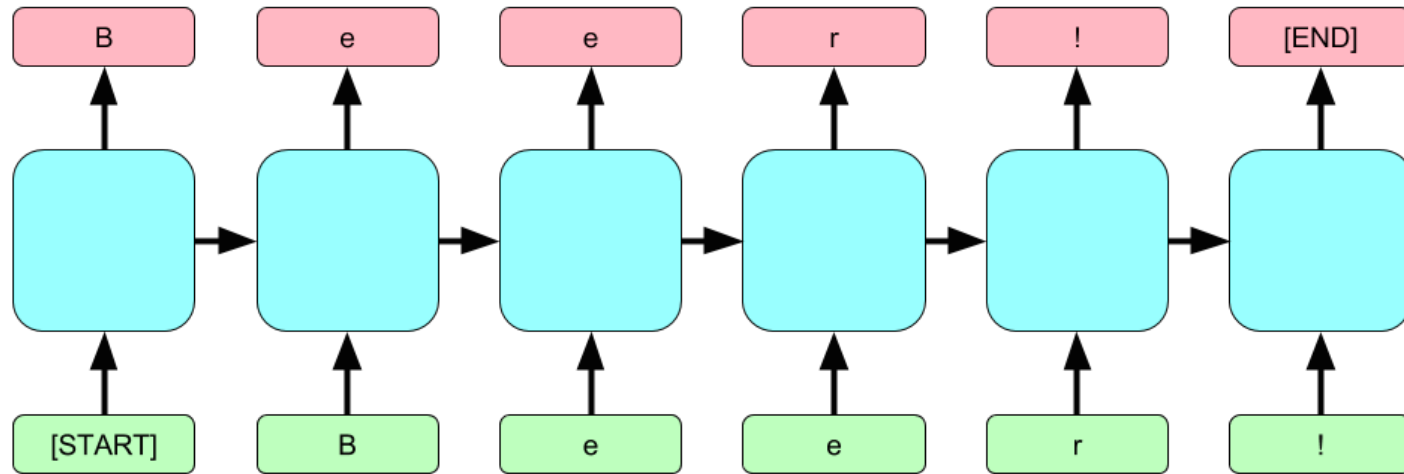
for all $x \in I_m$. In other words, functions of the form $F(x)$ are **dense** in $C(I_m)$.

At the cost of many parameters
and much difficulty to fit

Neural Networks



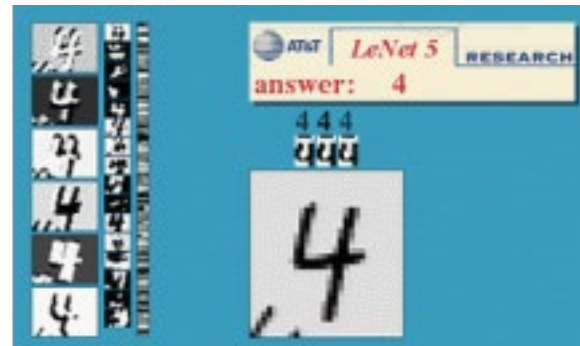
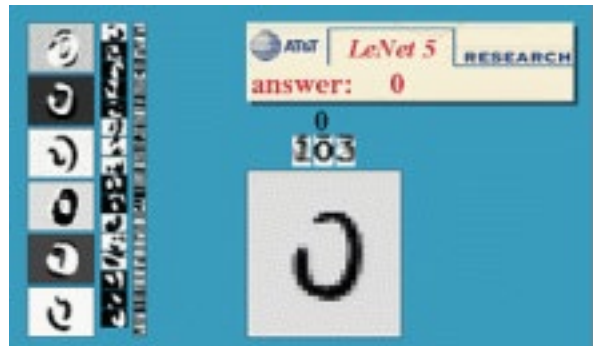
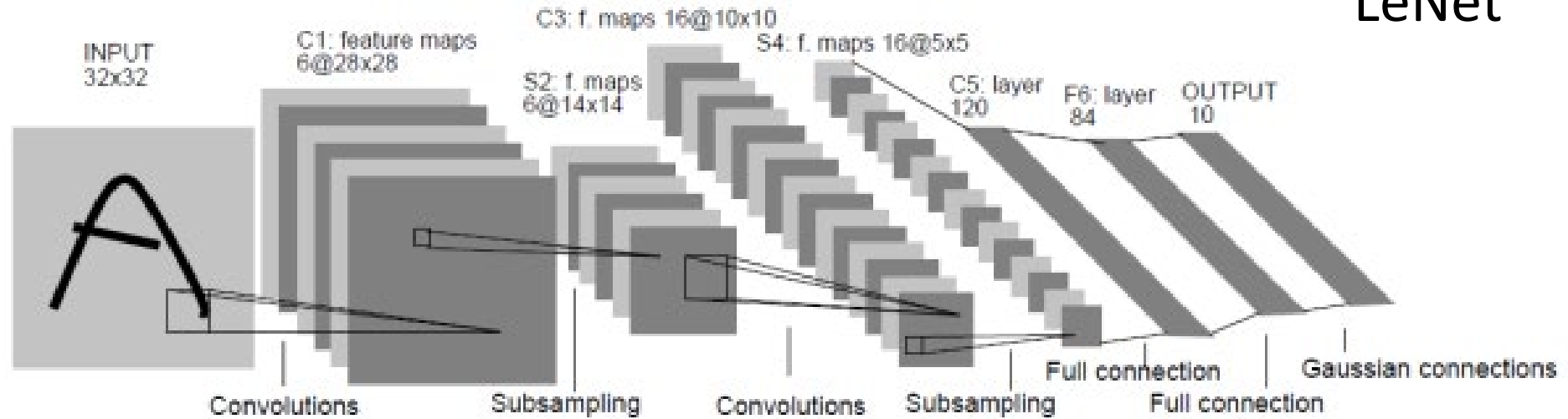
CNN



RNN

Convolutional Neural Networks

LeNet



One of the first successful applications of CNN.

Convolutional Neural Networks

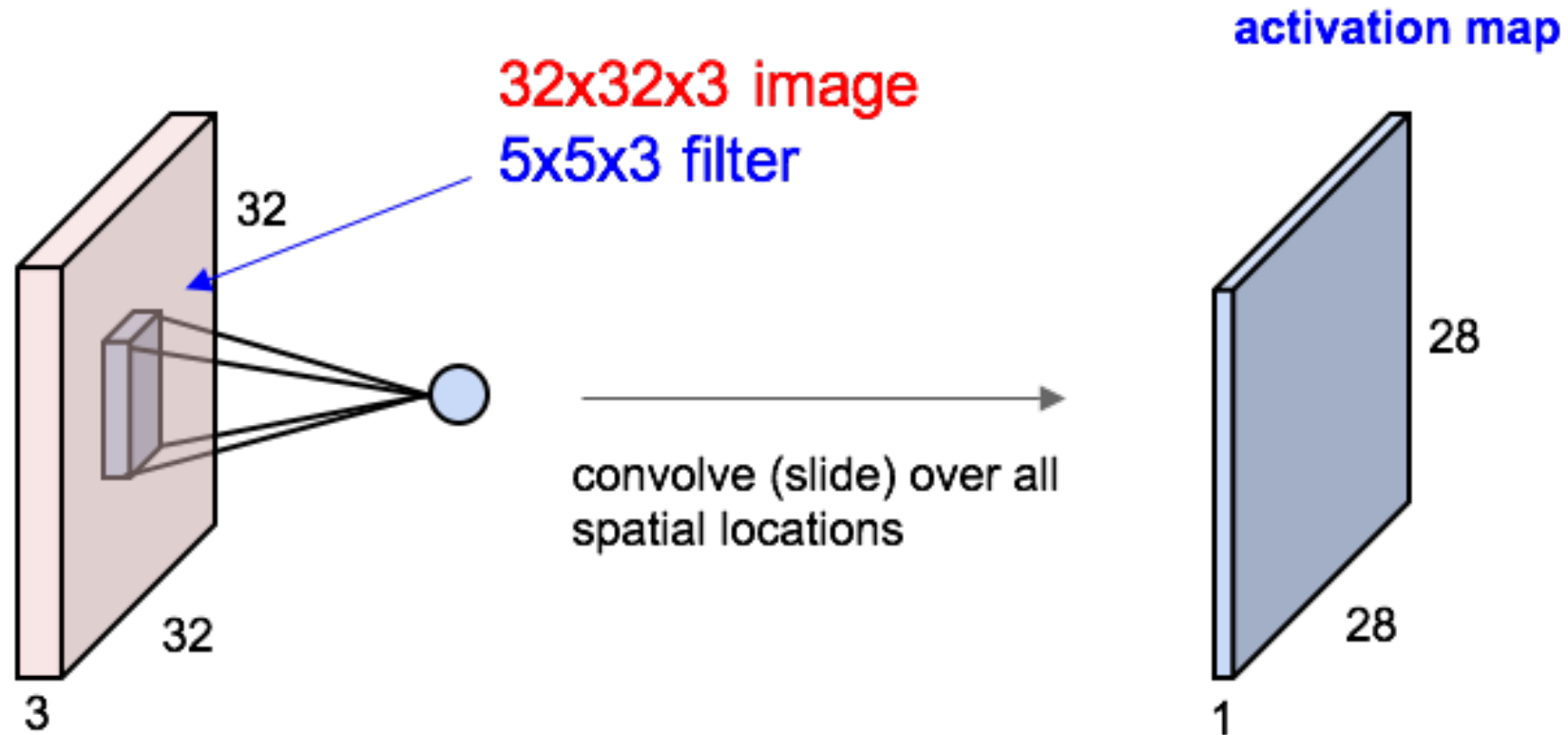


Image Credits: Andrej Karpathy

Convolutional Neural Networks

- ◆ Filters are doing pattern matching



Image Credits: Yan LeCun

ImageNet Challenge

ImageNet Dataset

IMAGENET

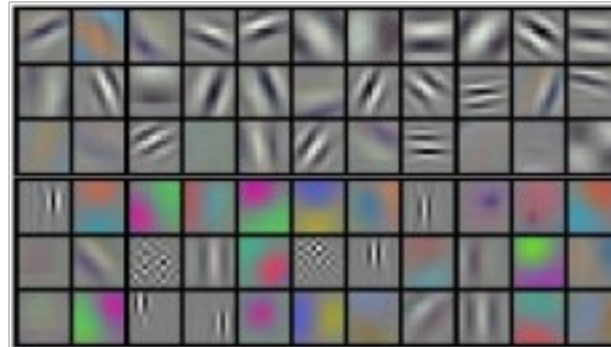
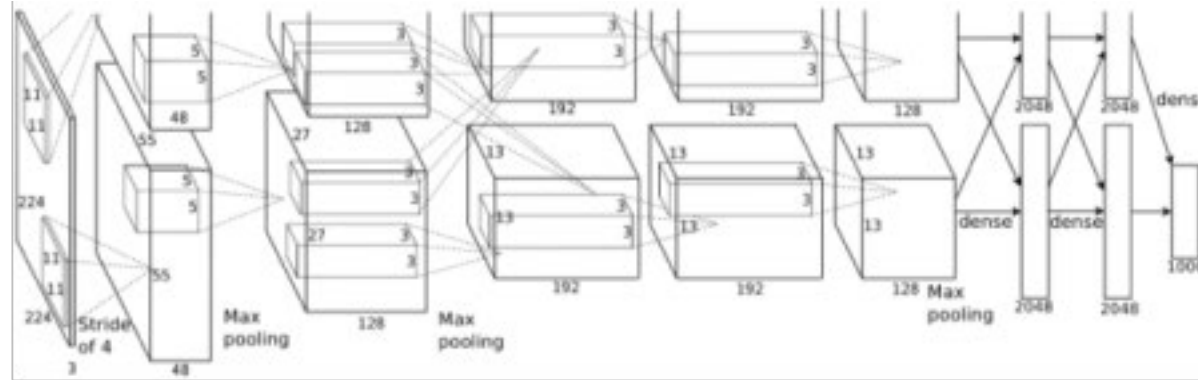
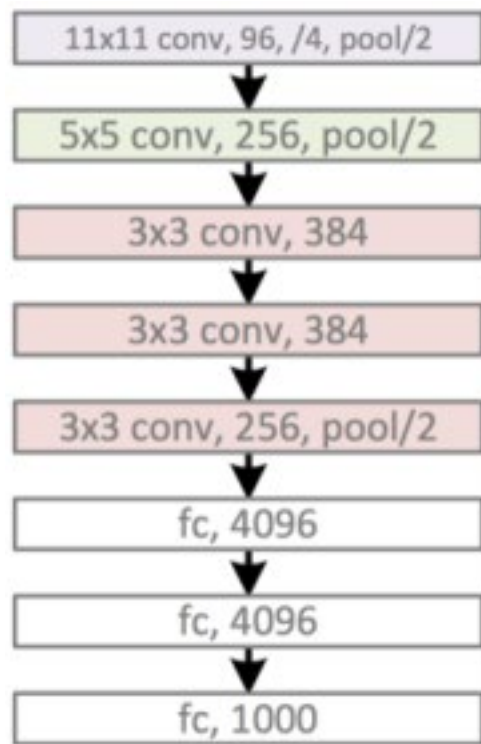


Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang et al. ["Imagenet large scale visual recognition challenge."](#) International Journal of Computer Vision 115, no. 3 (2015): 211-252. [\[web\]](#)

3

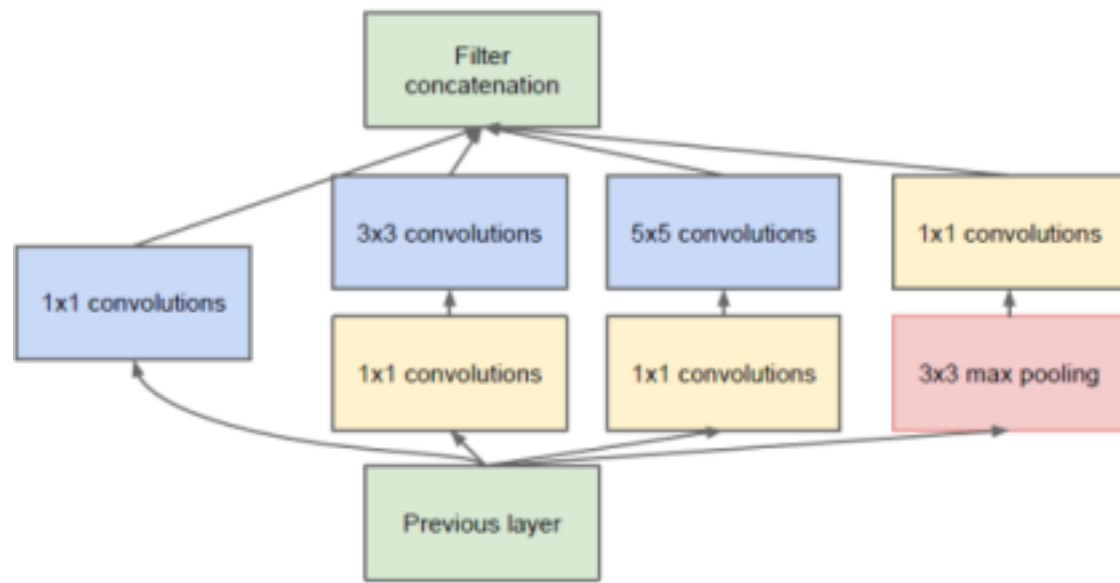
Convolutional Neural Networks

AlexNet



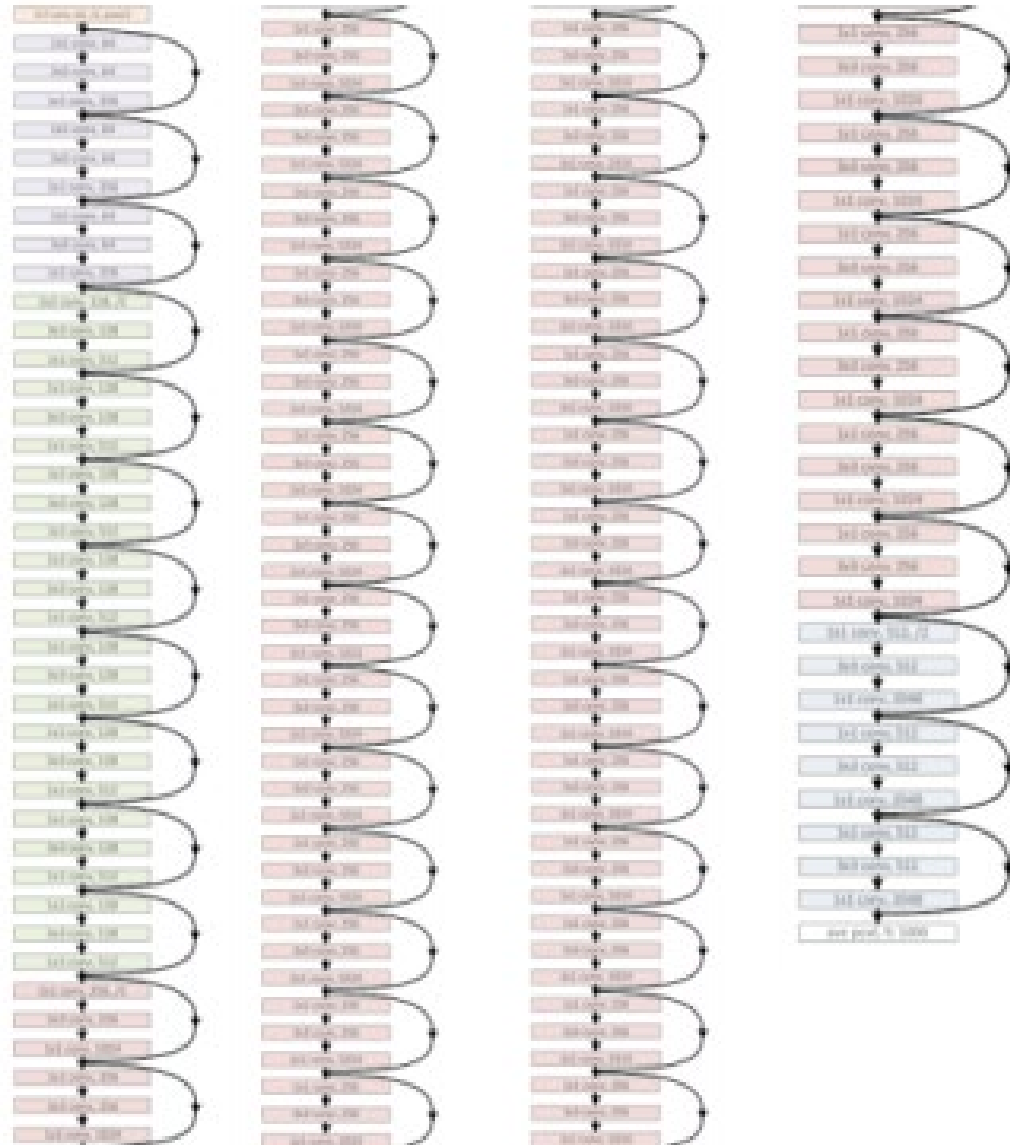
The first work that popularized Convolutional Networks in Computer Vision

Convolutional Neural Networks

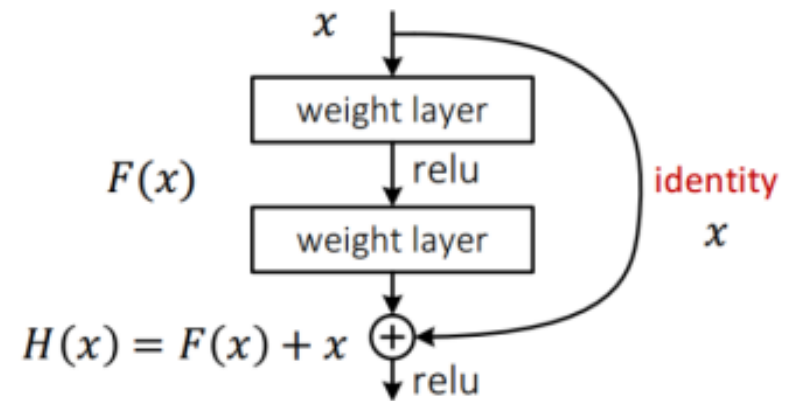


An Inception Module: a new building block..

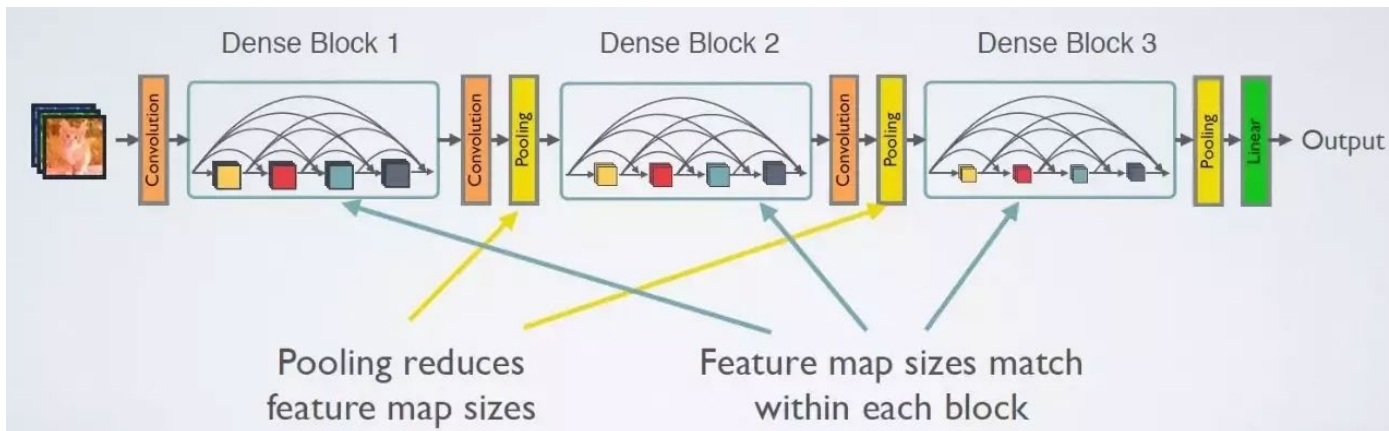
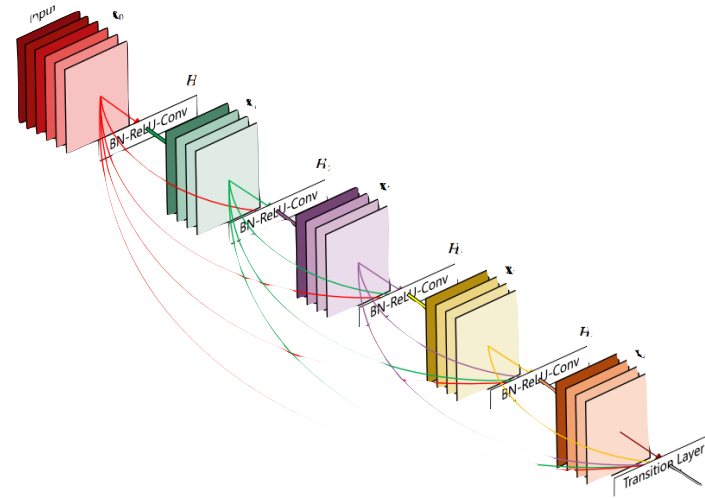
Convolutional Neural Networks



ResNet



Convolutional Neural Networks



DenseNet

DenseNet-264

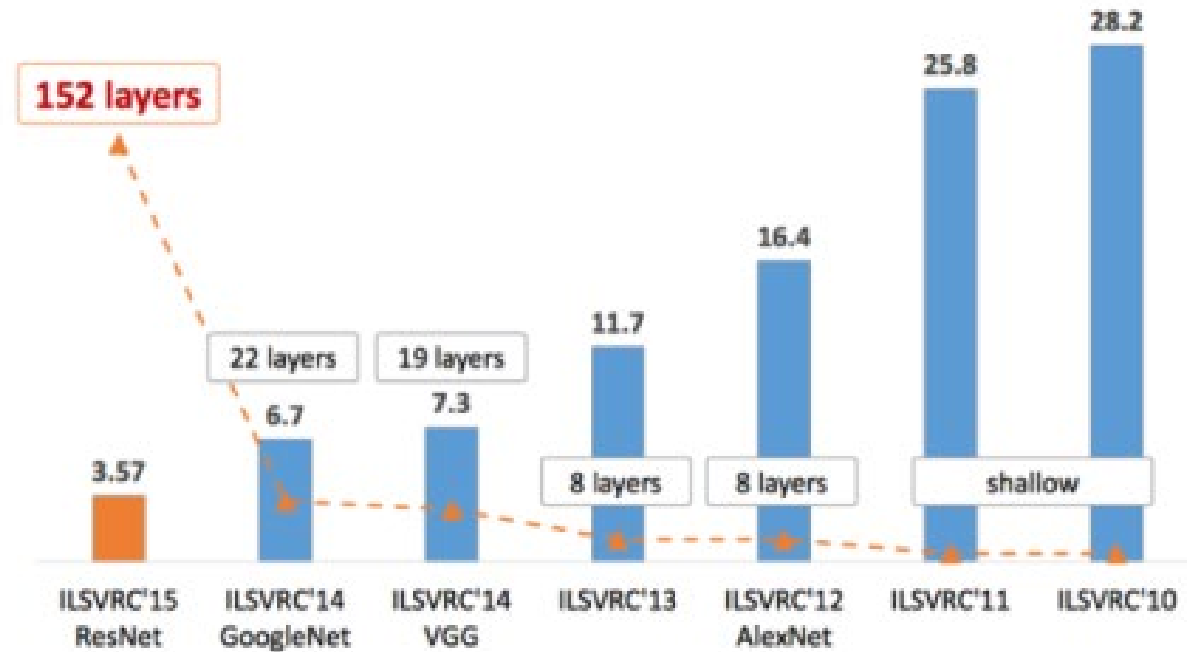
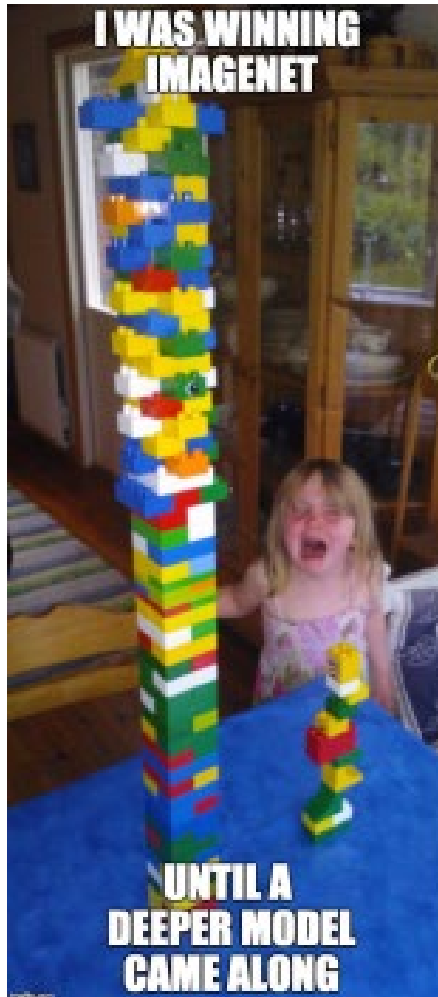
$$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$$

$$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$$

$$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 64$$

$$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$$

Convolutional Neural Networks



Classification Error: the lower, the better

Neural Networks as Feature Extractors

◆ How do neural networks work?

Neural Networks as Feature Extractors

- ◆ Traditional methods hand-engineer features

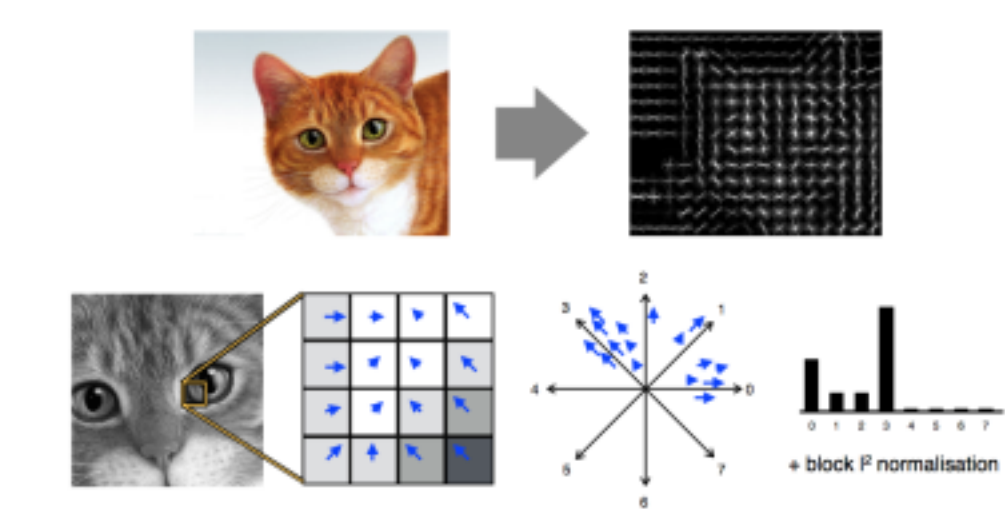
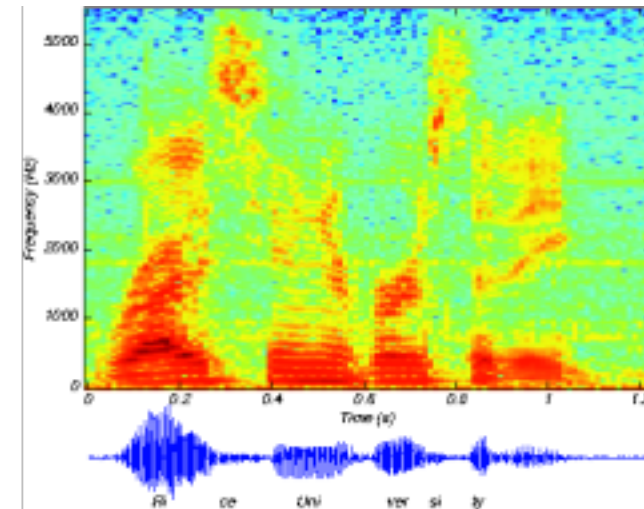


Image: HoG



Audio: Spectrogram

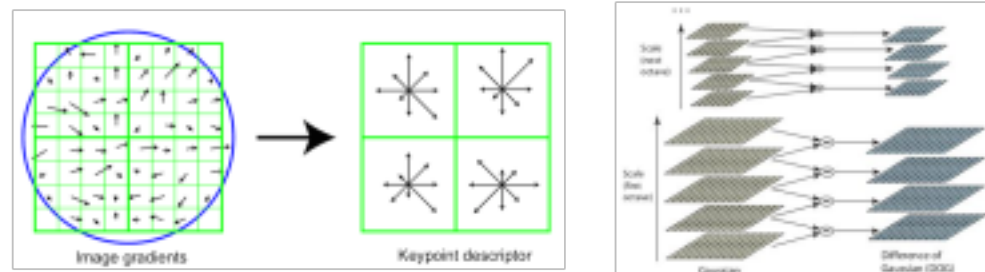
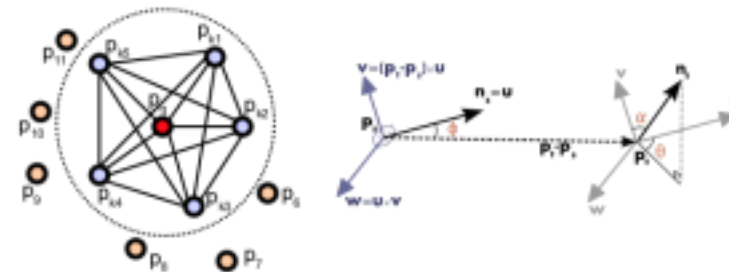


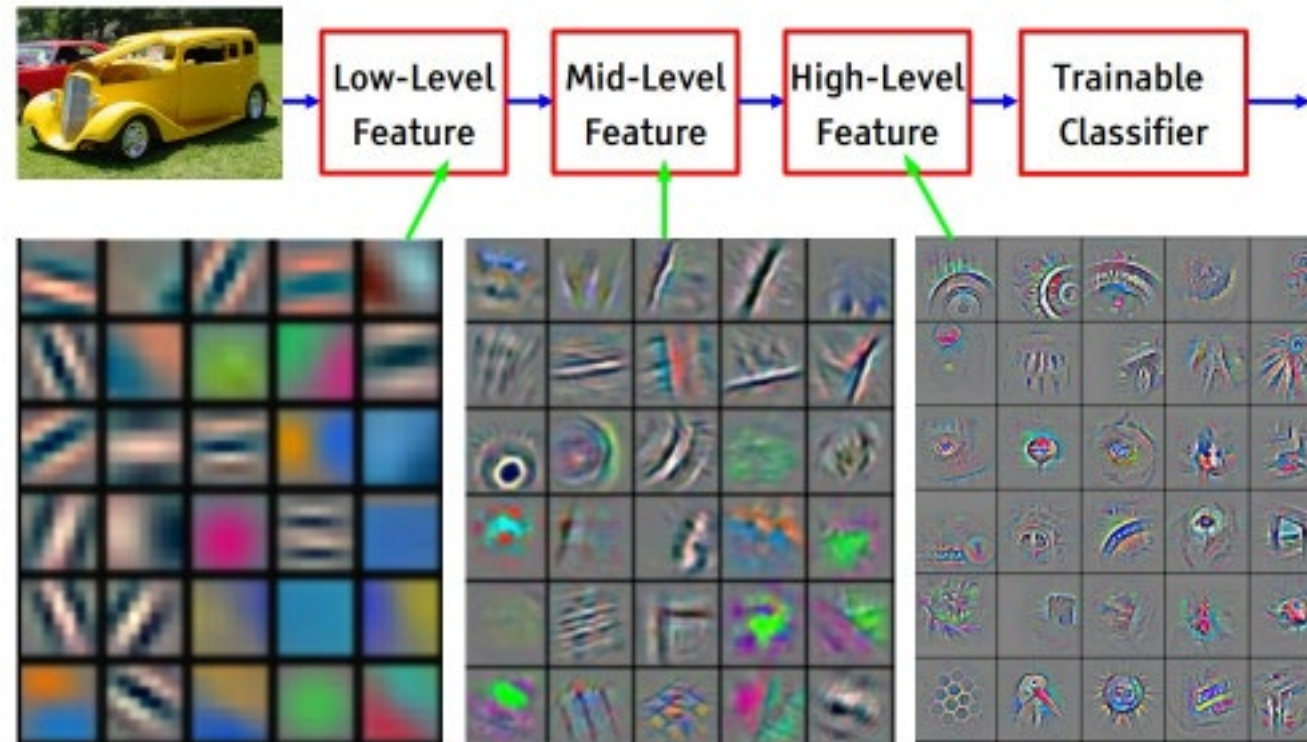
Image: SIFT



Point Cloud: PFH

Neural Networks as Feature Extractors

- ◆ Neural networks extract powerful features from data

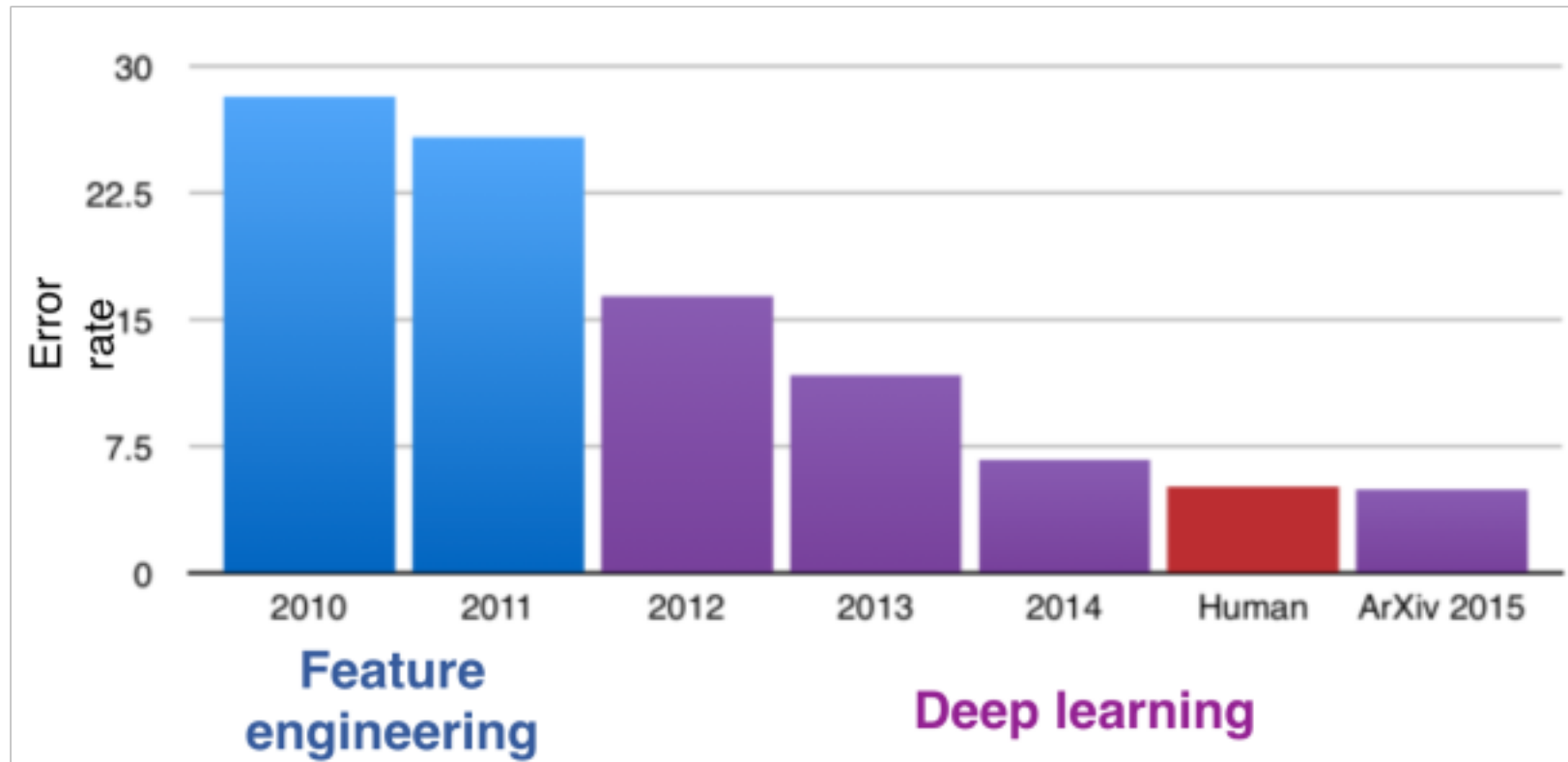


Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Image Credits: Yan LeCun

Neural Networks as Feature Extractors

ImageNet 1000 class image classification accuracy



Useful Courses

- ◆ CS 231n: <http://cs231n.stanford.edu/>
 - ◆ Deep Learning for Visual Data
- ◆ CS 224n: <http://cs224n.stanford.edu/>
 - ◆ Deep Learning for Language / Sequential Data
- ◆ CS 234: <http://cs234.stanford.edu/>
 - ◆ Reinforcement Learning

Programming Neural Networks

- ◆ TensorFlow, Python, Google
 - ◆ <https://www.tensorflow.org/>
- ◆ PyTorch, Python, Facebook
 - ◆ <https://pytorch.org/>
- ◆ Caffe, Berkeley
 - ◆ <http://caffe.berkeleyvision.org/>



Programming Neural Networks



```
y_train = keras.utils.to_categorical(y_train, num_classes)
y_test = keras.utils.to_categorical(y_test, num_classes)
```

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),
                activation='relu',
                input_shape=input_shape))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(num_classes, activation='softmax'))
```

```
model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adadelta(),
              metrics=['accuracy'])

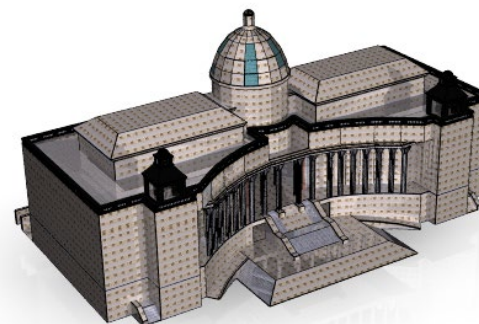
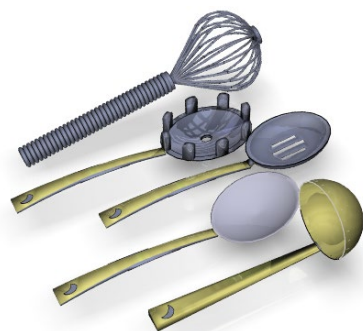
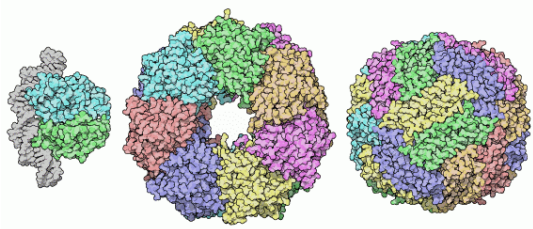
model.fit(x_train, y_train,
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,
          validation_data=(x_test, y_test))

score = model.evaluate(x_test, y_test, verbose=0)
print('Test loss:', score[0])
print('Test accuracy:', score[1])
```

Agenda

- ◆ Today
 - ◆ Deep Learning Intro
 - ◆ 3D Deep Learning
 - ◆ Multi-view CNNs
 - ◆ Volumetric CNNs

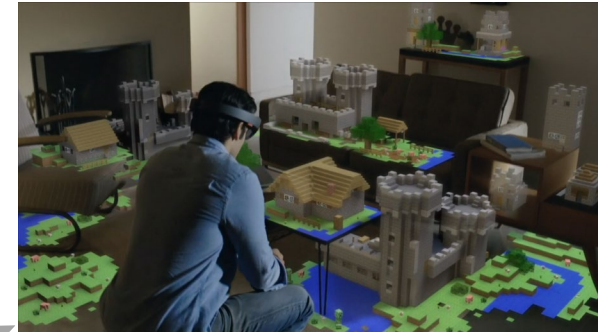
The World is 3D



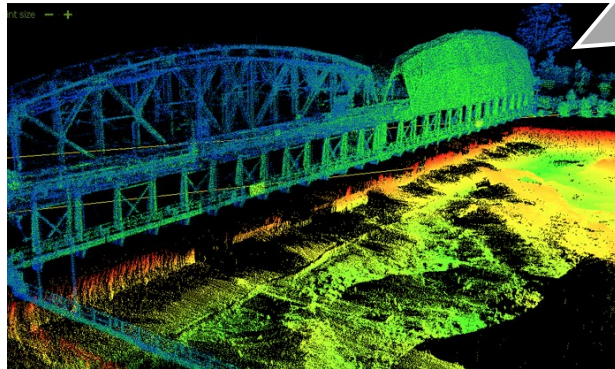
3D Applications



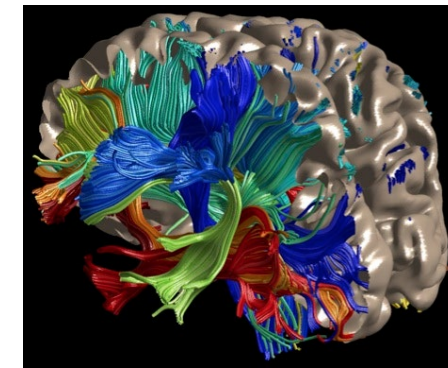
Robotics



Augmented Reality



Autonomous driving



Medical Image Processing

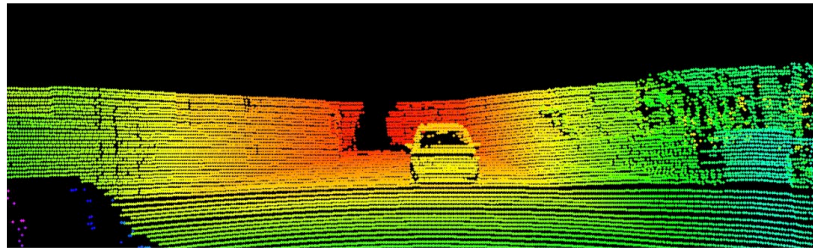
3D Perception is important for AI



Cosimo Alfredo Pina, "The domestic robots are getting closer"

Understanding in 3D is “Easier”!

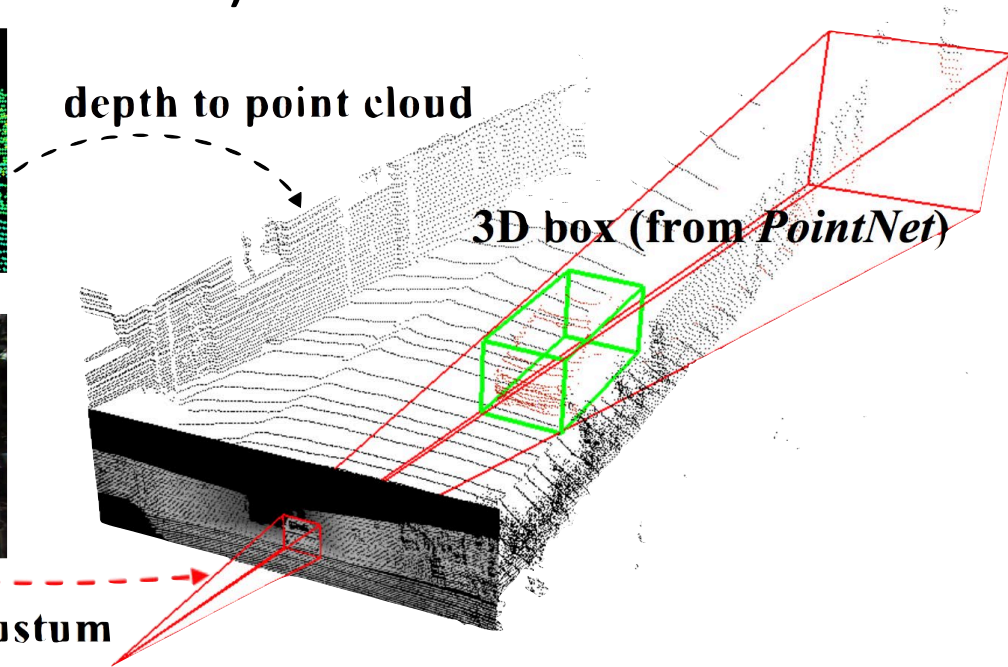
- ◆ Depth Data really helps (e.g. laser scan)



depth to point cloud

3D box (from *PointNet*)

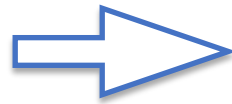
2D region (from *CNN*) to 3D frustum



Frustum PointNets for 3D Object Detection from RGB-D Data, Charles Qi et. al.

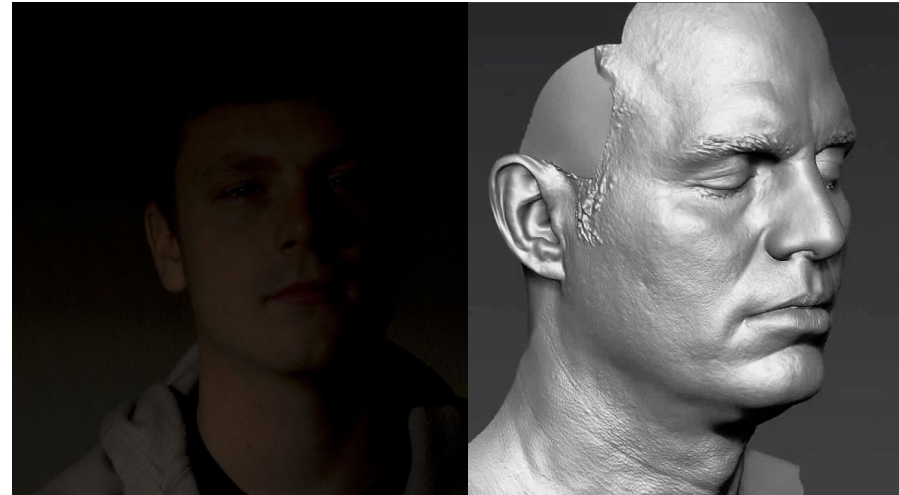
Understanding in 3D is “Easier”!

- ◆ 3D scanners even can give us full 3D models!



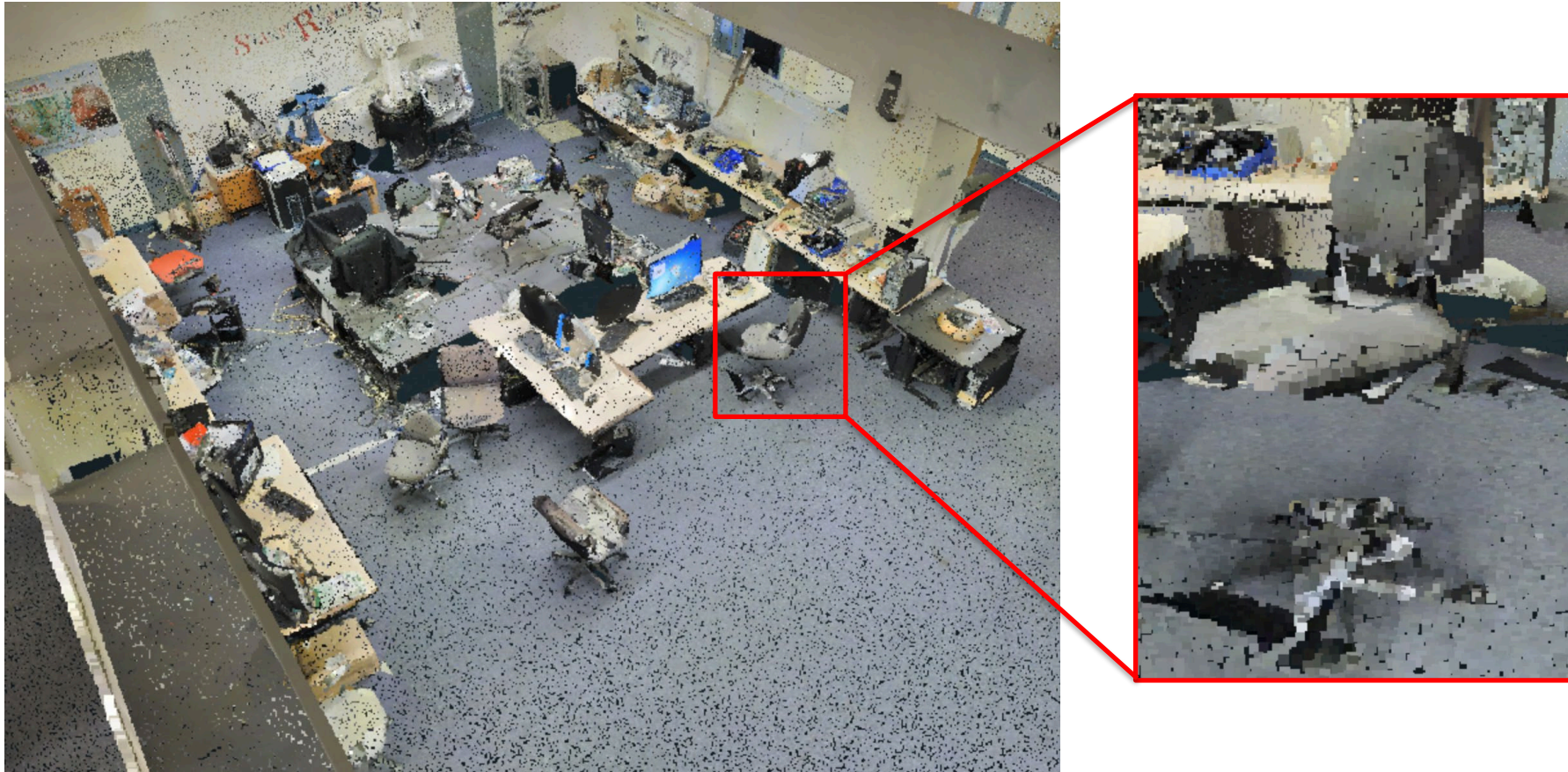
Understanding in 3D is “Easier”!

- ◆ 3D data is less vulnerable to different occlusions, different lighting conditions, etc.



Understanding in 3D is Harder!

- ◆ 3D data is often noisy and of low quality.



Understanding in 3D is Harder!

- ◆ 3D data is often incomplete.

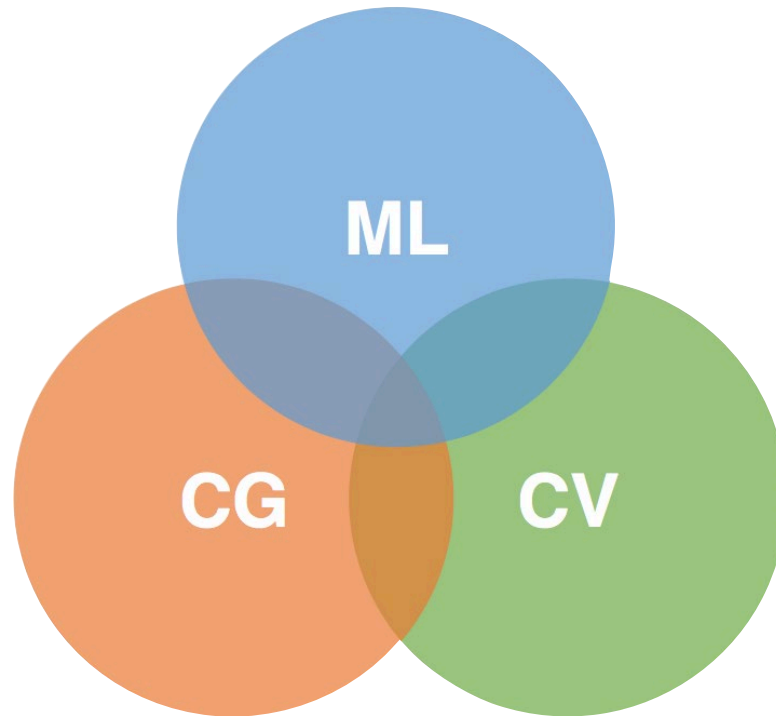


Understanding in 3D is Harder!

- ◆ 3D data is hard to acquire
 - Hard to Scan real-world models
 - Hard to design models in CAD softwares
- ◆ Compared to 2D images, 3D data is hard to acquire
- ◆ Less data to train NNs
- ◆ Harder to gain annotation labels

3D Deep Learning

- ◆ A field with very short history — starting from 2015
- ◆ But very active due to huge industry interests!



Big Data in 3D

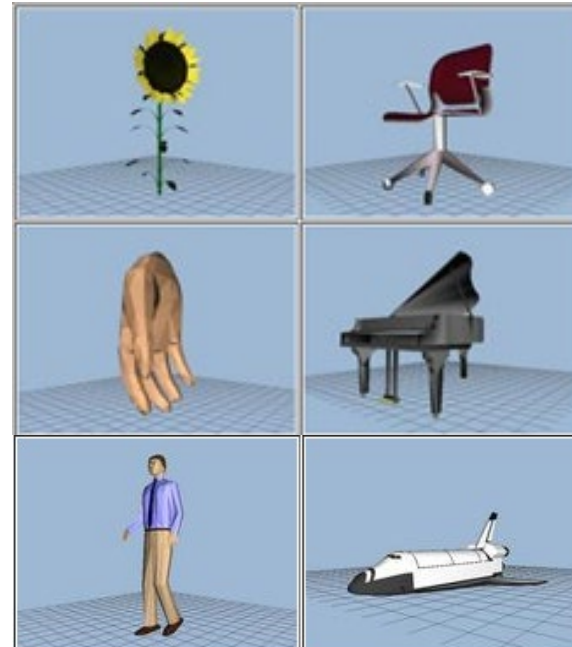
Status as of 2010:



Stanford bunny

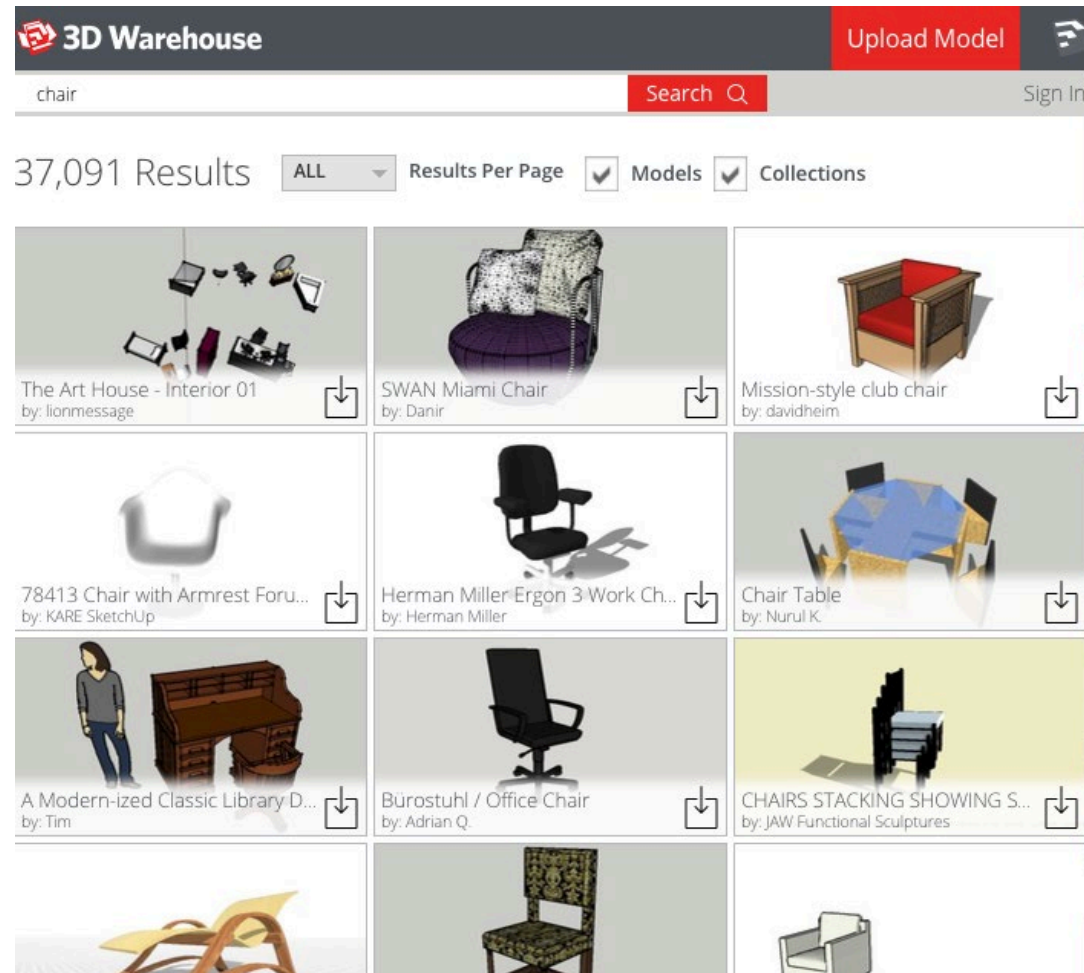


1800 models in 90 categories



Princeton shape benchmark
[Shilane et al. 04]

Big Data in 3D



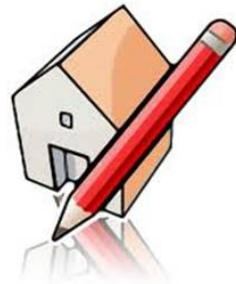
Nowadays millions of 3D models in online repositories

Big Data in 3D

Growing crowd-sourced market for 3D models



Clara.io



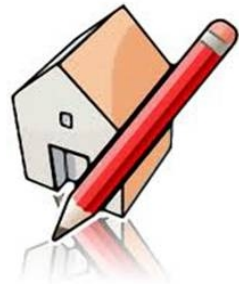
.....

Big Data in 3D

Growing crowd-sourced market for 3D models

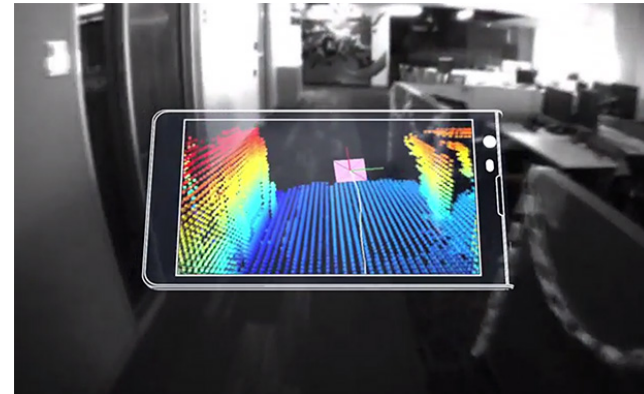


An opportunity for data-driven
3D visual computing



.....

Big Data in 3D



Big Data in 3D



airplane,aeroplane,plane

an aircraft that has a fixed wing and is powered by propellers or jets; 'the flight was delayed due to trouble with the airplane'

[ImageNet](#) [MetaData](#)

Choose a taxonomy:




























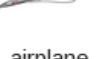

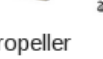


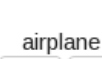





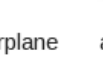

ShapeNetCore ▾

- airplane,aeroplane,plane(11,4045)
- ashcan,trash can,garbage can,wastebin,...
- bag,traveling bag,traveling bag,grip,suitc...
- basket,handbasket(2,113)
- bathtub,bathing tub,bath,tub(0,856)
- bed(13,233)
- bench(5,1813)
- bicycle,bike,wheel,cycle(0,59)
- birdhouse(0,73)
- bookshelf(0,452)
- bottle(6,498)
- bowl(1,186)
- bus,autobus,coach,charabanc,double-de...
- cabinet(9,1571)
- camera,photographic camera(4,113)
- can,tin,tin can(2,108)
- cap(4,56)
- car,auto,automobile,machine,motorcar(18...
- chair(23,6778)
- clock(3,651)
- computer keyboard,keypad(0,65)
- dishwasher,dish washer,dishwashing ma...

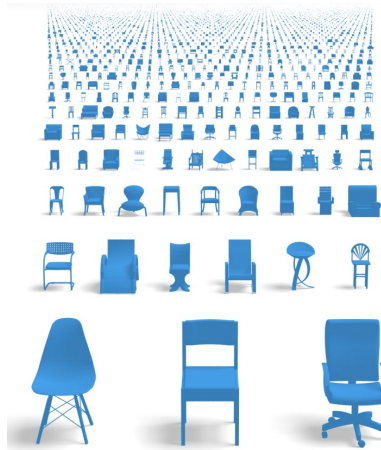
Synset Models TreeMap Stats

Displaying 1 to 160 of 4045

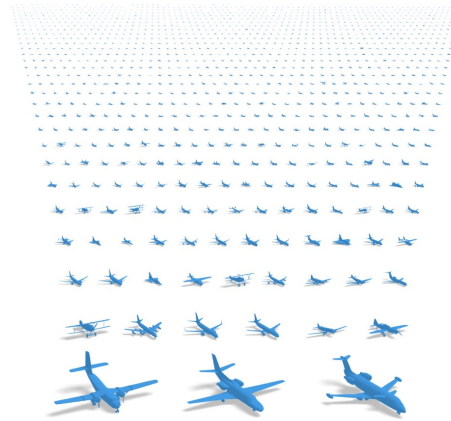
< 1 2 3 4 5 6 7 8 9 10 11 ... 26 >

							
airplane	airplane	airplane	airplane	bomber	fighter	airliner	straight wing
							
airplane	airplane	airplane	propeller plane	airplane	airplane	airplane	bomber
							
airplane	airplane	airliner	delta wing	airplane	airplane	jet	jet
							
airplane	airplane	airplane	airplane	airplane	propeller plane	airplane	airplane
							
airplane	jet	airliner	fighter	fighter	airplane	airplane	airplane

ShapeNet: A Large-scale 3D Model Database

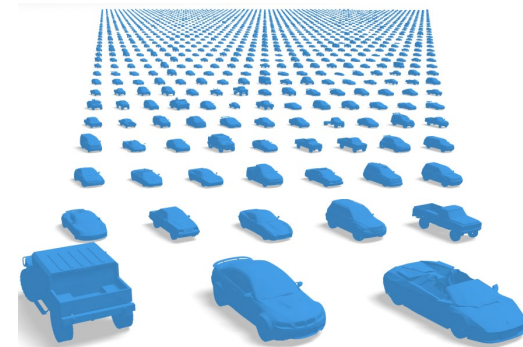


~3 million models in total



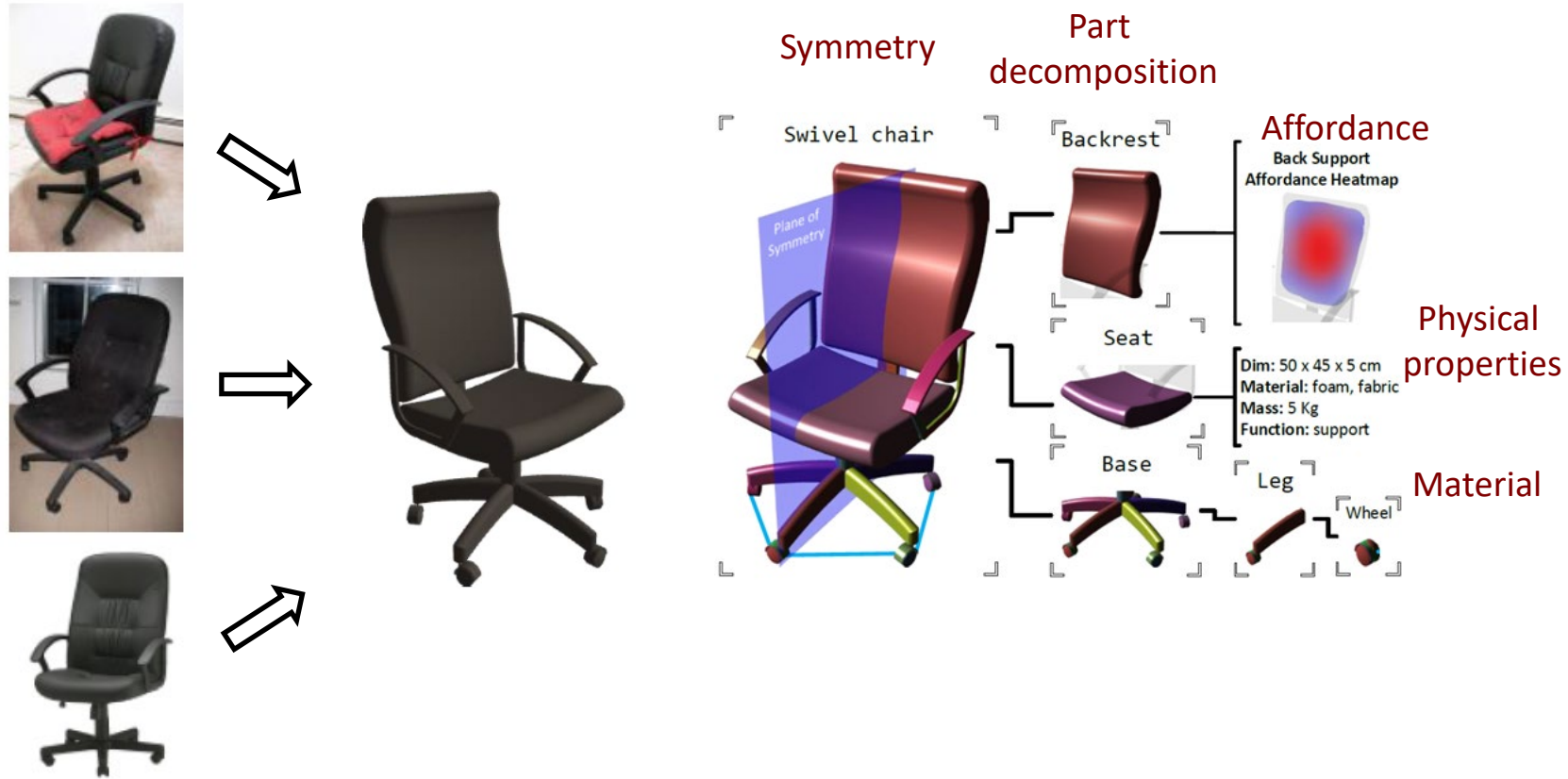
~2,000 classes

...



Rich annotations

Unified Knowledge Representation in 3D



3D Deep Learning Tasks

3D geometry analysis



Classification



Parsing
(object/scene)



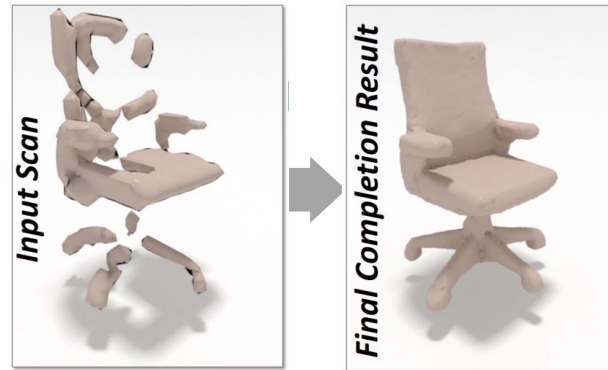
Correspondence

3D Deep Learning Tasks

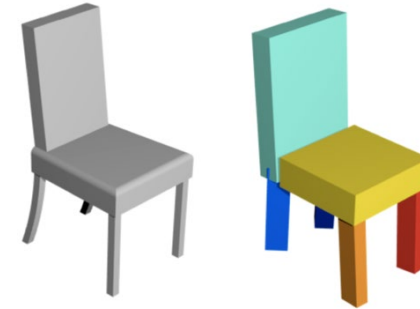
3D Synthesis



Monocular
3D reconstruction



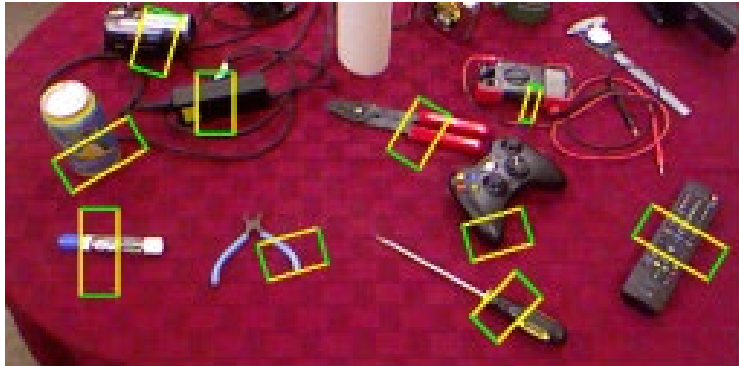
Shape completion



Shape modeling

3D Deep Learning Tasks

Robotics Applications



Robot Grasping Affordance Map

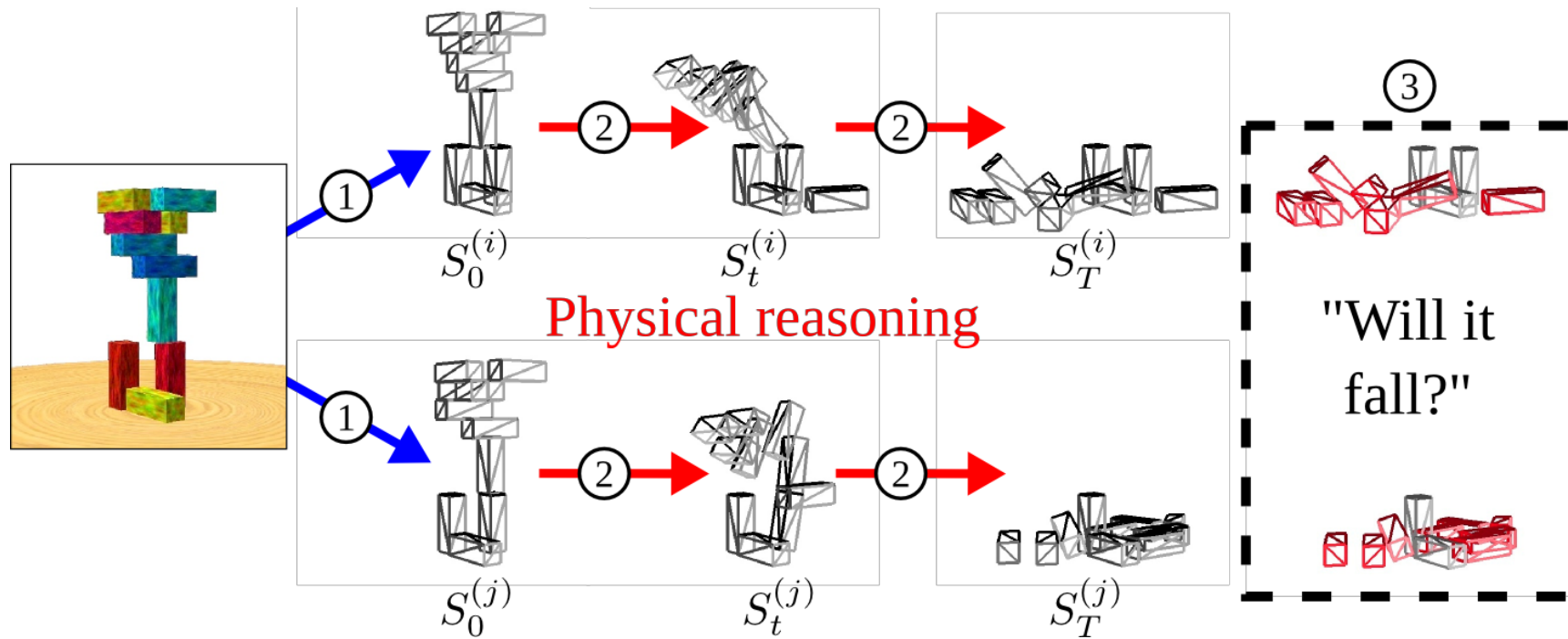


Human Interaction Affordance Map



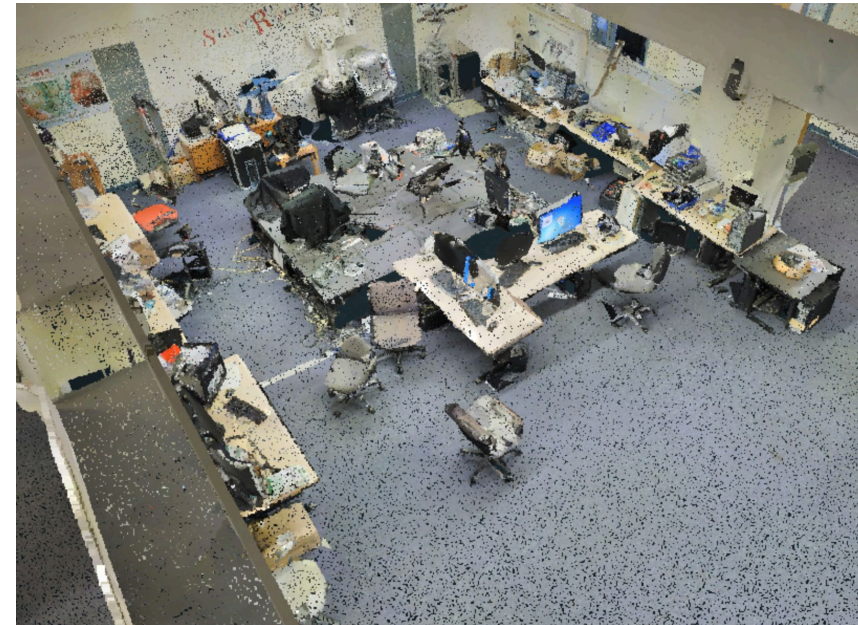
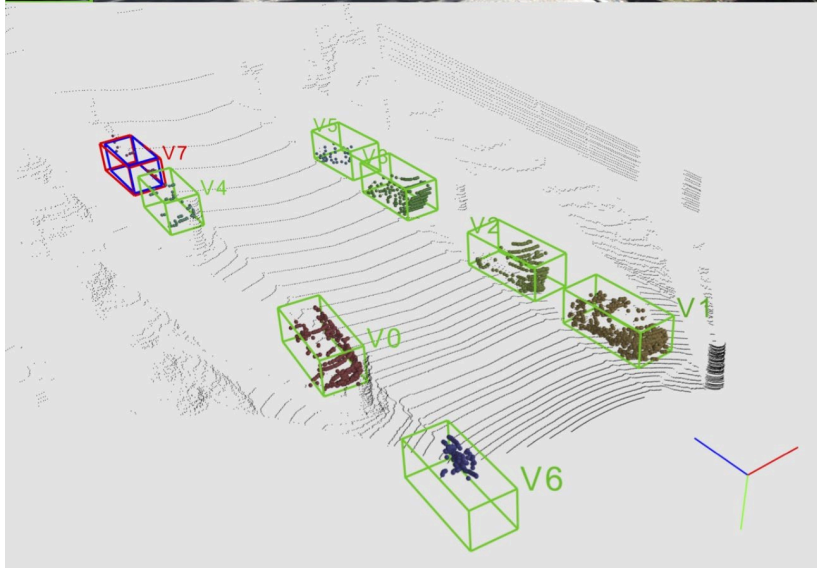
3D Deep Learning Tasks

Intuitive Physics based on 3D Understanding



3D Representation

◆ How to represent 3D data?



2D Representation

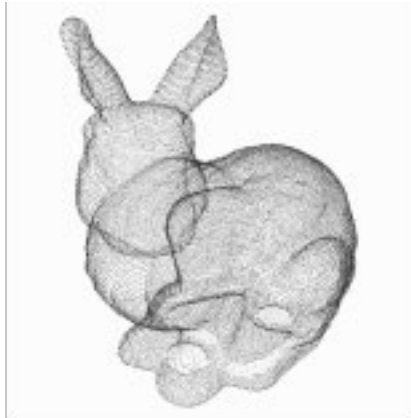
Images: canonical representation with regular data structure



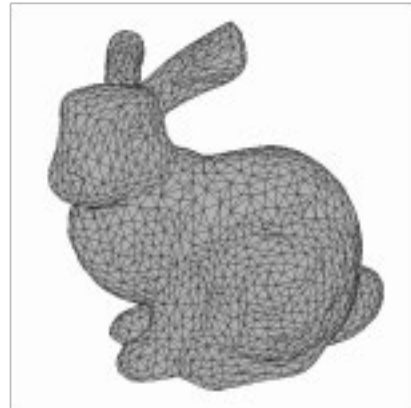
1	44	33	12	20	23	35	14
51	16	40	32	46	48	28	17
29	60	3	63	49	55	36	7
52	22	26	41	38	10	61	53
2	24	19	11	34	43	5	8
57	9	37	42	25	21	27	18
30	56	50	64	4	59	6	13
58	47	45	31	39	15	62	54

3D Representation

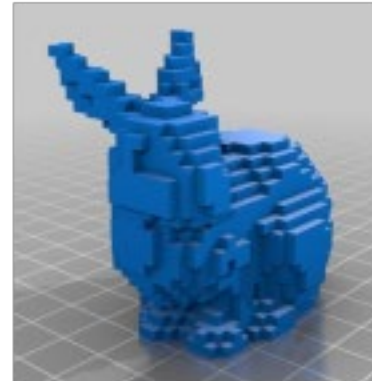
- ◆ Design good 3D representation for NN to consume



Point Cloud



Mesh



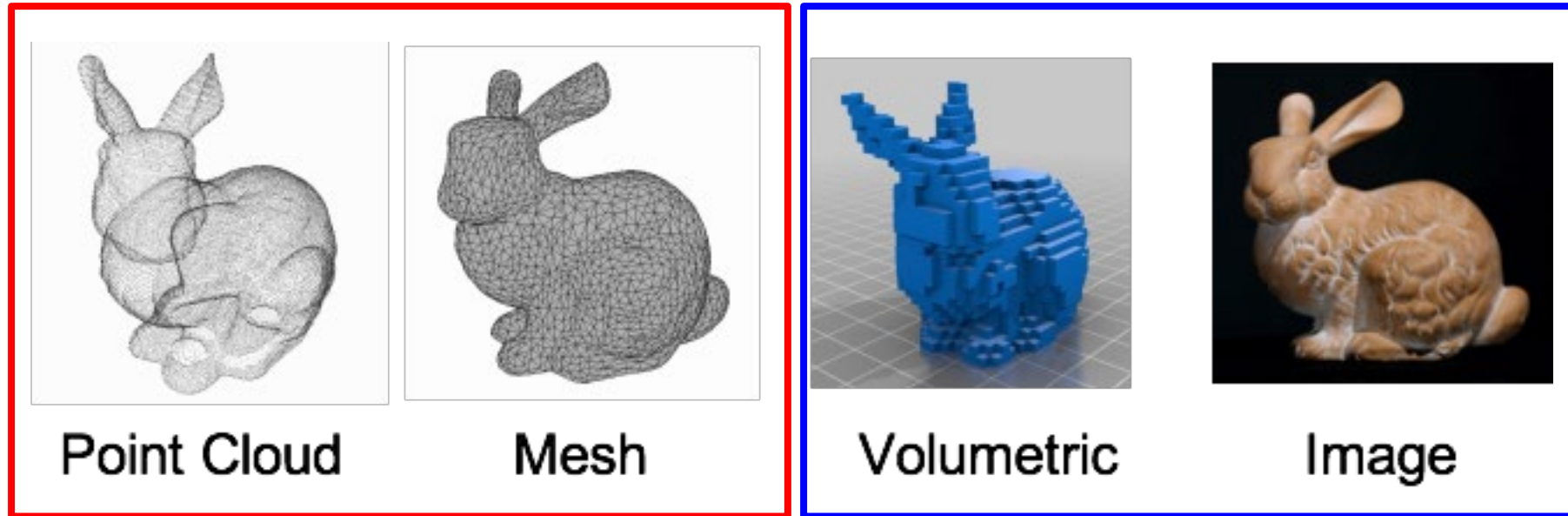
Volumetric



Image

3D Representation

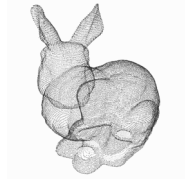
- ◆ Design good 3D representation for NN to consume



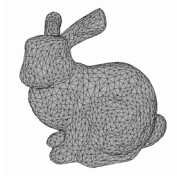
Irregular

Regular

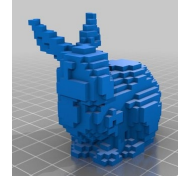
3D Representation



Point Cloud



Mesh

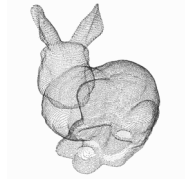


Volumetric

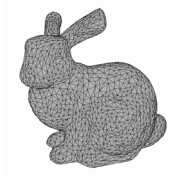


Image

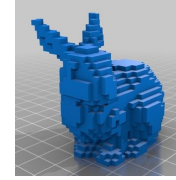
3D Representation



Point Cloud



Mesh



Volumetric



Image

Rawness

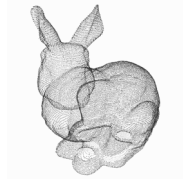
High

Low

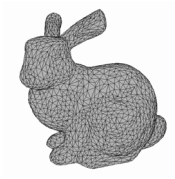
Medium

High

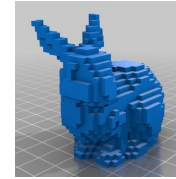
3D Representation



Point Cloud



Mesh



Volumetric



Image

Rawness

High

Low

Medium

High

**3D
Geometry**

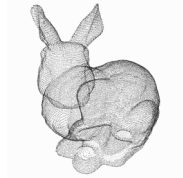
Yes

Yes

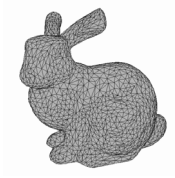
Yes

No

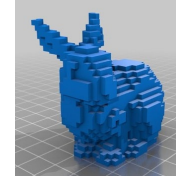
3D Representation



Point Cloud



Mesh



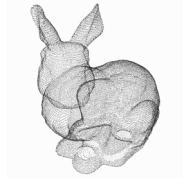
Volumetric



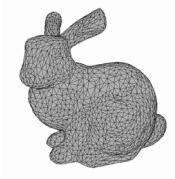
Image

	Point Cloud	Mesh	Volumetric	Image
Rawness	High	Low	Medium	High
3D Geometry	Yes	Yes	Yes	No
Compactness	Yes	Yes	No	Medium

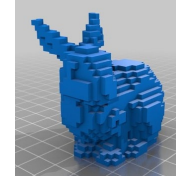
3D Representation



Point Cloud



Mesh



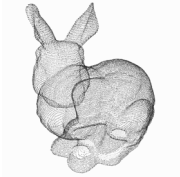
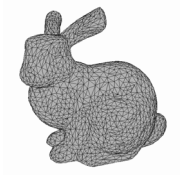
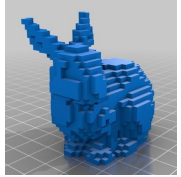

Volumetric



Image

	Point Cloud	Mesh	Volumetric	Image
Rawness	High	Low	Medium	High
3D Geometry	Yes	Yes	Yes	No
Compactness	Yes	Yes	No	Medium
Data Structure	Set { X_i }	Graph (V, E)	Array X_{ijk}	Array X_{ij}

3D Representation

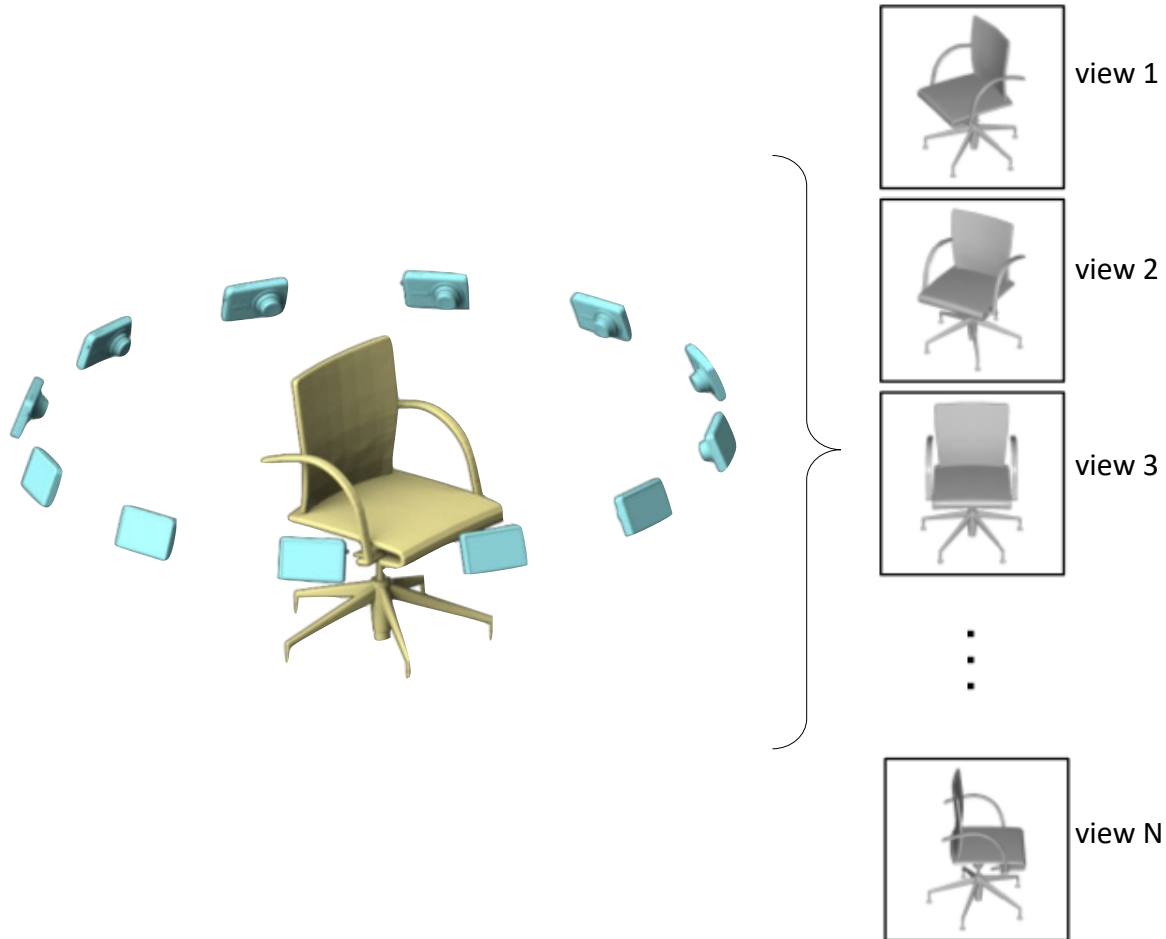
	 Point Cloud	 Mesh	 Volumetric	 Image
Rawness	High	Low	Medium	High
3D Geometry	Yes	Yes	Yes	No
Compactness	Yes	Yes	No	Medium
Data Structure	Set $\{X_i\}$	Graph (V, E)	Array X_{ijk}	Array X_{ij}

Again, no free lunch!

Agenda

- ◆ Today
 - ◆ Deep Learning Intro
 - ◆ 3D Deep Learning
 - ◆ **Multi-view CNNs**
 - ◆ Volumetric CNNs

Multi-view Representation



+ Powerful
2D CNNs

Multi-view Representation

- ◆ Pros

- ◆ Regular grids representation
- ◆ Easy to obtain the input images
- ◆ Easy to input to Neural Nets
- ◆ Leverage state-of-the-art 2D CNNs

- ◆ Cons

- ◆ Background clutters / lightings / etc.
- ◆ No access to the original 3D content
- ◆ Each view only contains partial information and aggregation is needed

Multi-view Representation

- ◆ Classification
- ◆ Segmentation



This is a chair!



Multi-view CNNs: Classification



This is a chair!

Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-
Miller, "**Multi-view Convolutional Neural Networks for 3D
Shape Recognition**", *Proceedings of ICCV 2015*

Image Credits: Hang Su

Multi-view CNNs: Classification



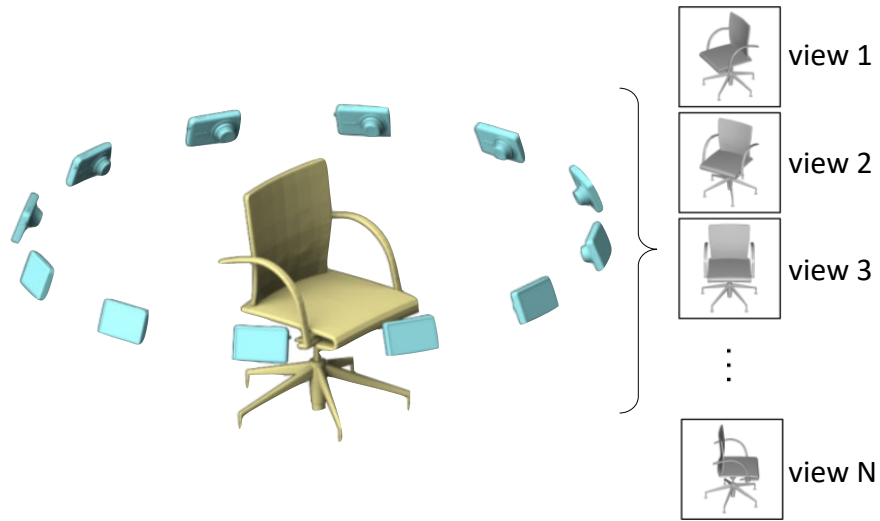
Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-

Miller, "**Multi-view Convolutional Neural Networks for 3D**

Shape Recognition", *Proceedings of ICCV 2015*

Image Credits: Hang Su

Multi-view CNNs: Classification



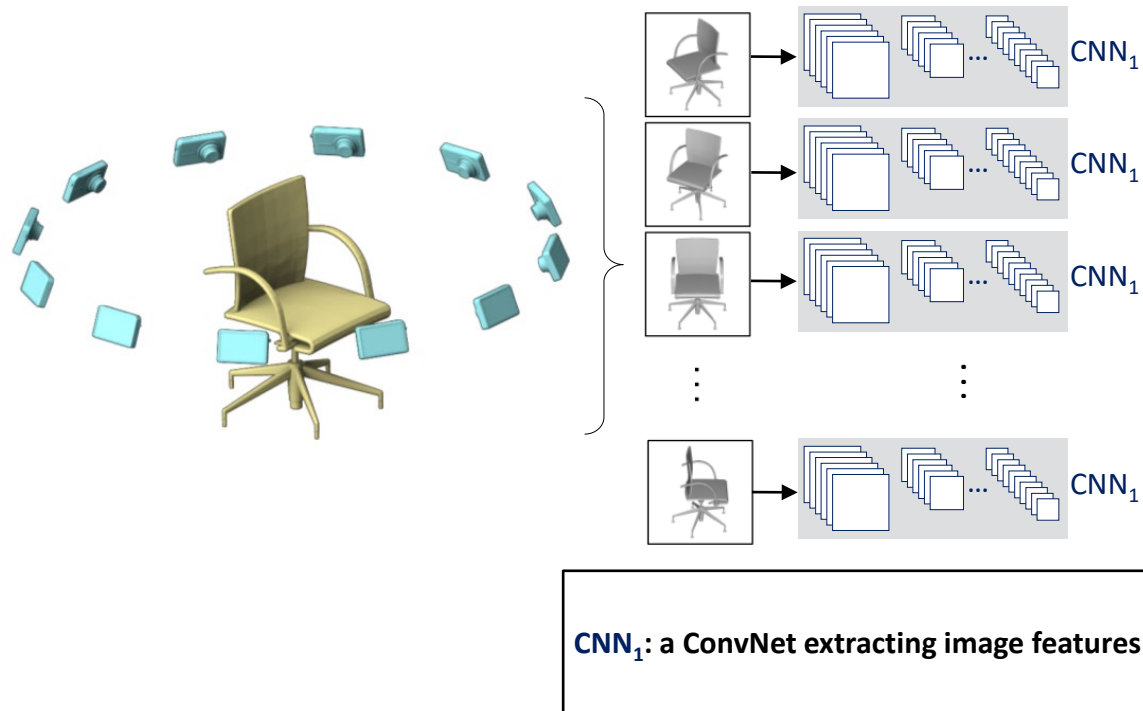
Hang Su, Subhansu Maji, Evangelos Kalogerakis, Erik Learned-

Miller, "**Multi-view Convolutional Neural Networks for 3D**

Shape Recognition", *Proceedings of ICCV 2015*

Image Credits: Hang Su

Multi-view CNNs: Classification



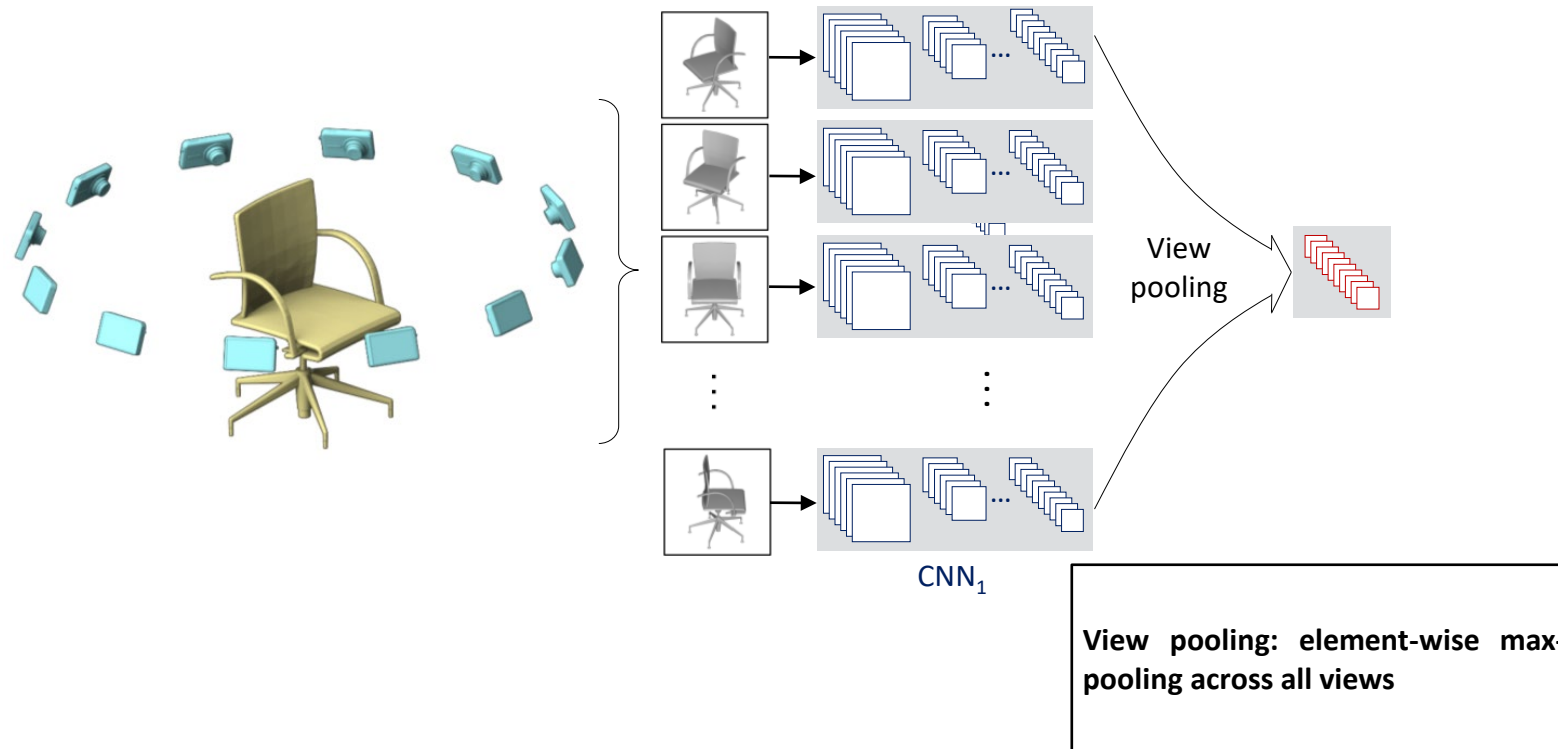
Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-

Miller, "Multi-view Convolutional Neural Networks for 3D

Shape Recognition", *Proceedings of ICCV 2015*

Image Credits: Hang Su

Multi-view CNNs: Classification



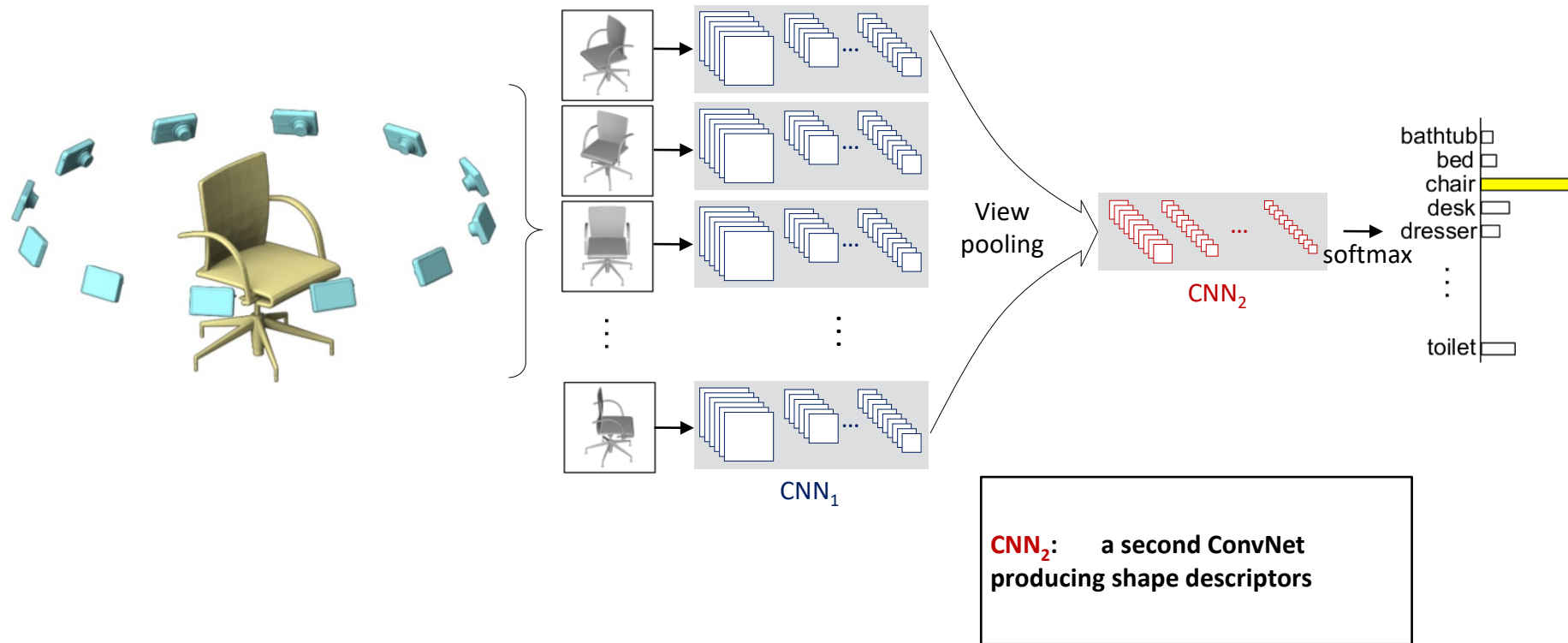
Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-

Miller, "Multi-view Convolutional Neural Networks for 3D

Shape Recognition", *Proceedings of ICCV 2015*

Image Credits: Hang Su

Multi-view CNNs: Classification



Hang Su, Subhransu Maji, Evangelos Kalogerakis, Erik Learned-

Miller, "Multi-view Convolutional Neural Networks for 3D

Shape Recognition", *Proceedings of ICCV 2015*

Image Credits: Hang Su

Multi-view CNNs: Classification

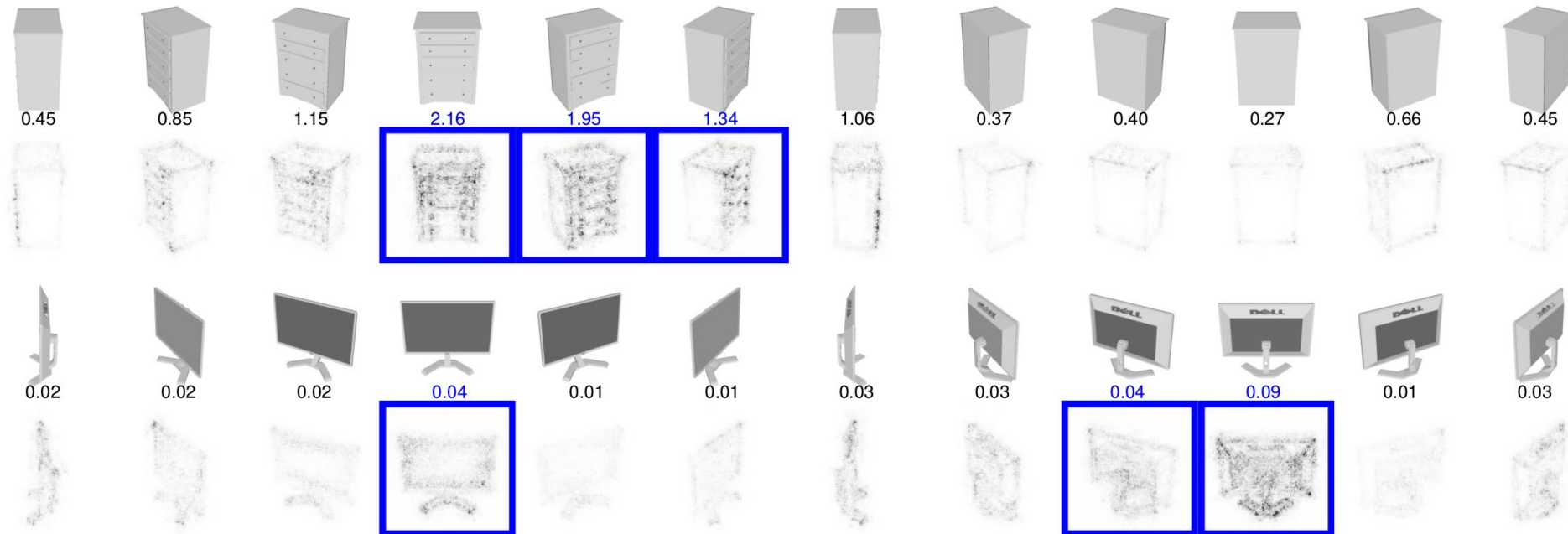
On **ModelNet40**, compared against:

- 3 existing methods:
SPH, LFD, 3D ShapeNets
- 2 strong baselines:
Fisher vectors, CNN

Method	Classification (Accuracy)	Retrieval (mAP)
SPH [16]	68.2%	33.3%
LFD [5]	75.5%	40.9%
3D ShapeNets [37]	77.3%	49.2%
FV, 12 views	84.8%	43.9%
CNN, 12 views	88.6%	62.8%
MVCNN, 12 views	89.9%	70.1%
MVCNN+metric, 12 views	89.5%	80.2%
MVCNN, 80 views	90.1%	70.4%
MVCNN+metric, 80 views	90.1%	79.5%

Multi-view CNNs: Classification

$$[\omega_1, \omega_2 \dots \omega_K] = \left[\frac{\partial F_c}{\partial I_1} \Big|_S, \frac{\partial F_c}{\partial I_2} \Big|_S \dots \frac{\partial F_c}{\partial I_K} \Big|_S \right]$$



Multi-view CNNs: Segmentation

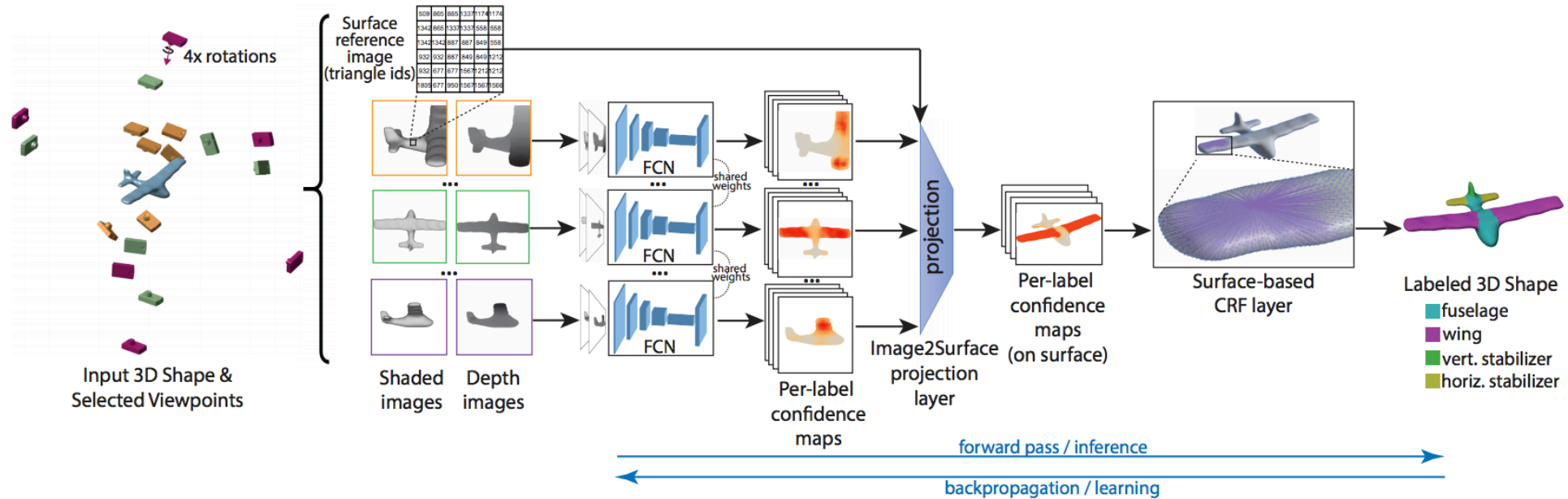


Image Credits: Eric Yi

Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyuan Yan, Hao Su, Cewu Lu, Qixing

Huang, Alla Sheffer, Leonidas J. Guibas, "A Scalable Active Framework for Region Annotation in

Multi-view CNNs: Segmentation

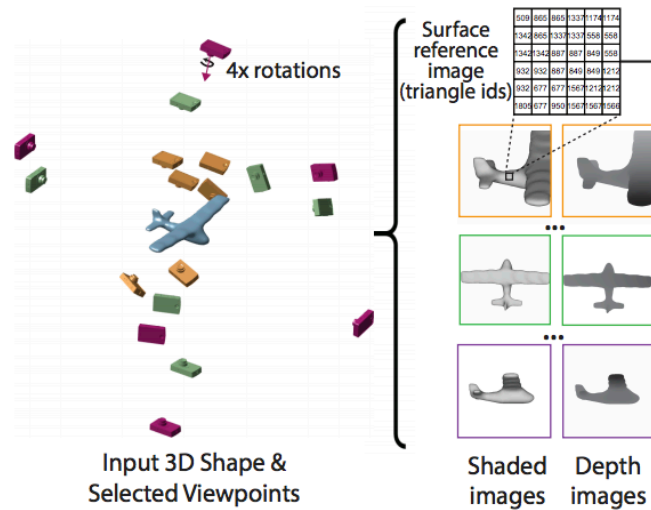


Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, Siddhartha Chaudhuri,

“3D Shape Segmentation with Projective Convolutional Networks”,

CVPR2017

Multi-view CNNs: Segmentation



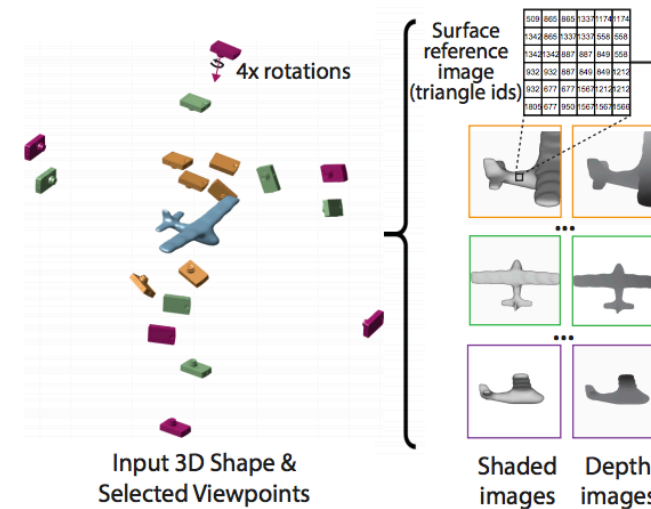
Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, Siddhartha Chaudhuri,

“3D Shape Segmentation with Projective Convolutional Networks”,

CVPR2017

Multi-view CNNs: Segmentation

- ◆ Parts have different sizes
 - ◆ Sample from different distances
- ◆ Make sure all points are visible
 - ◆ Compute points coverage
- ◆ Make the boundaries sharp
 - ◆ Use depth images

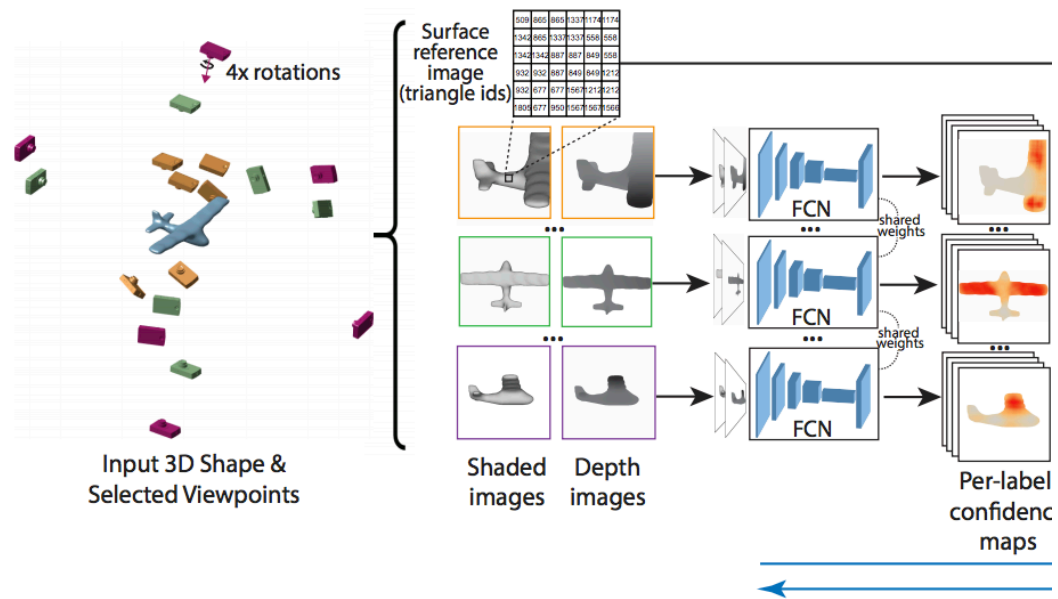


Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, Siddhartha Chaudhuri,

“3D Shape Segmentation with Projective Convolutional Networks”,

CVPR2017

Multi-view CNNs: Segmentation



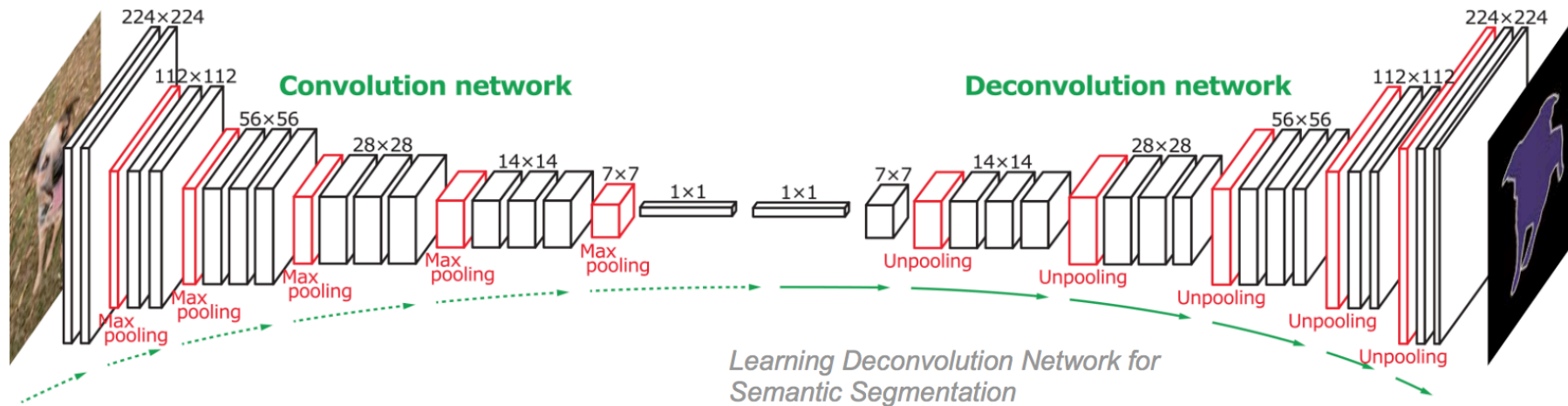
Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, Siddhartha Chaudhuri,

“3D Shape Segmentation with Projective Convolutional Networks”,

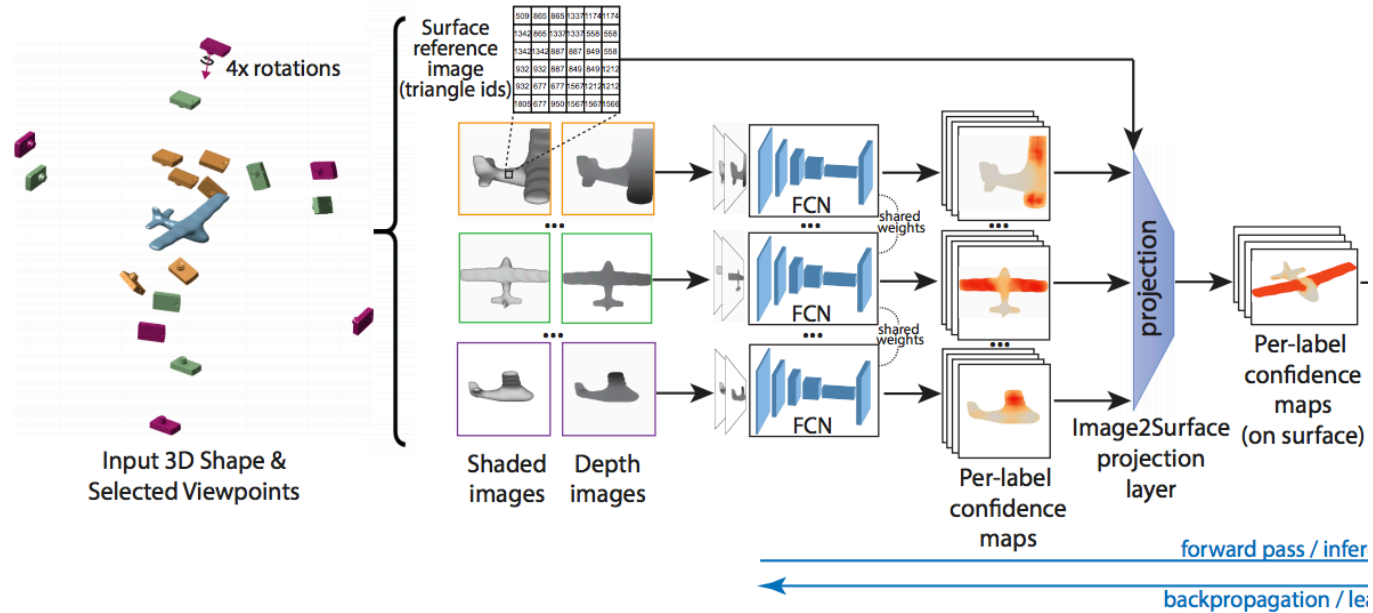
CVPR2017

Fully Convolutional Network

Segmentation:



Multi-view CNNs: Segmentation

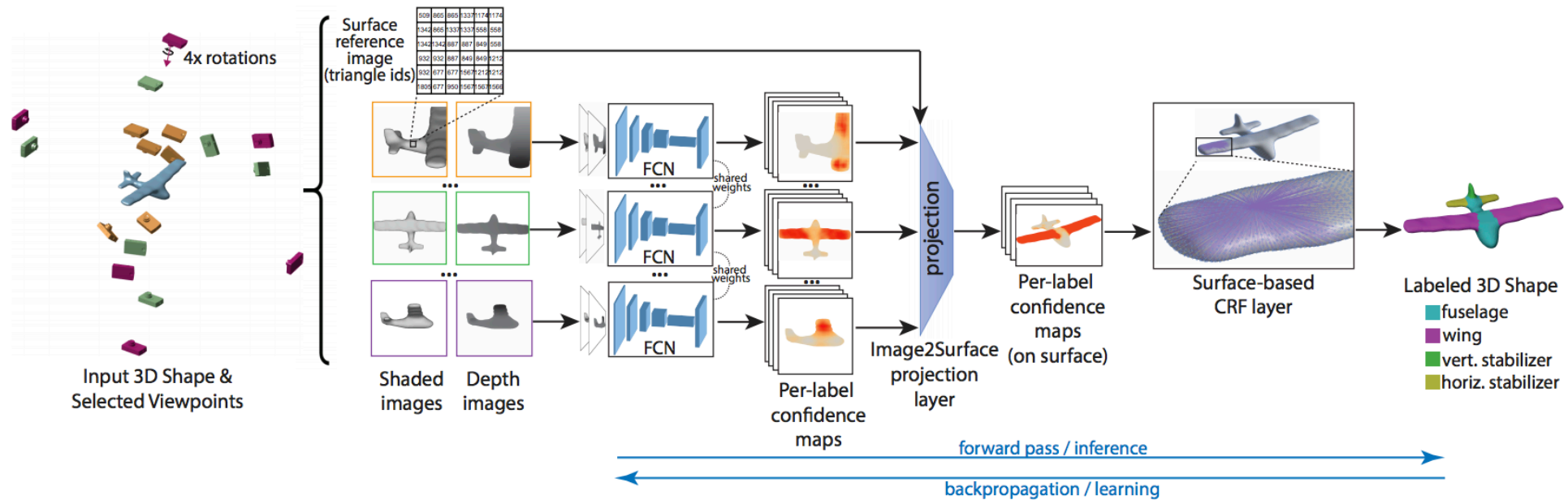


Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, Siddhartha Chaudhuri,

“3D Shape Segmentation with Projective Convolutional Networks”,

CVPR2017

Multi-view CNNs: Segmentation

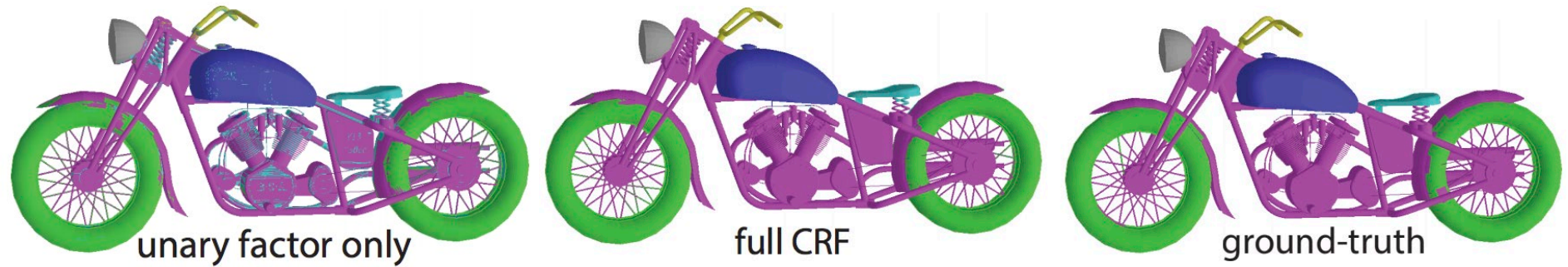


Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, Siddhartha Chaudhuri,

“3D Shape Segmentation with Projective Convolutional Networks”,

CVPR2017

Multi-view CNNs: Segmentation



Evangelos Kalogerakis, Melinos Averkiou, Subhransu Maji, Siddhartha Chaudhuri,

“3D Shape Segmentation with Projective Convolutional Networks”,

CVPR2017

Multi-view Representation Issues

- ◆ Cannot process invisible points
- ◆ View-based representations have redundancy
- ◆ Aggregation of view representations via max-pooling may lose information
- ◆ Properly fusing information across viewpoints is not incorporated in the network (not trivial).

Agenda

- ◆ Today
 - ◆ Deep Learning Intro
 - ◆ 3D Deep Learning
 - ◆ Multi-view CNNs
 - ◆ Volumetric CNNs

Volumetric Representation

Images: canonical representation with regular data structure



1	44	33	12	20	23	35	14
51	16	40	32	46	48	28	17
29	60	3	63	49	55	36	7
52	22	26	41	38	10	61	53
2	24	19	11	34	43	5	8
57	9	37	42	25	21	27	18
30	56	50	64	4	59	6	13
58	47	45	31	39	15	62	54

Volumetric Representation

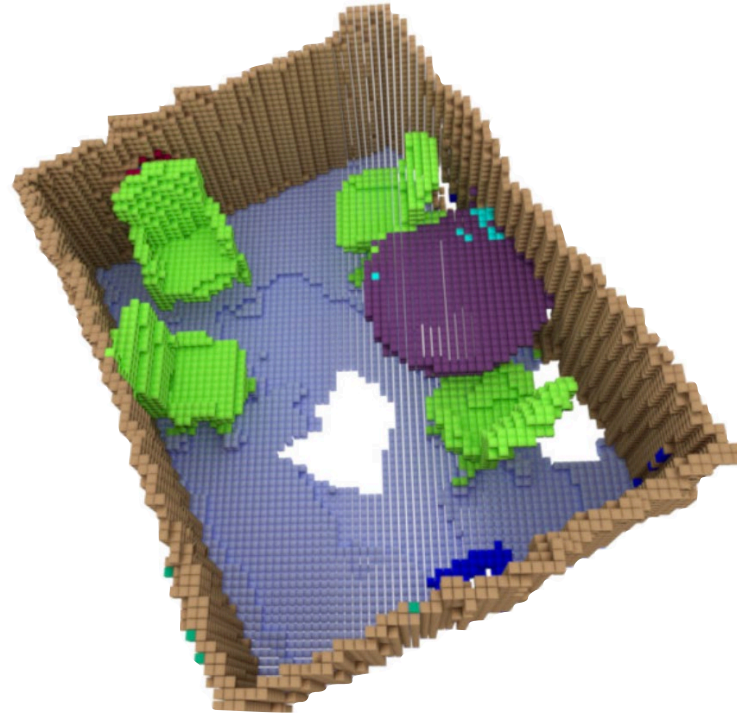
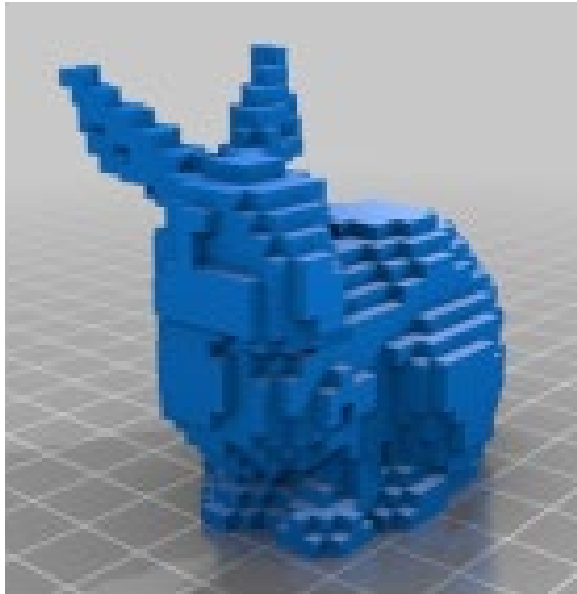
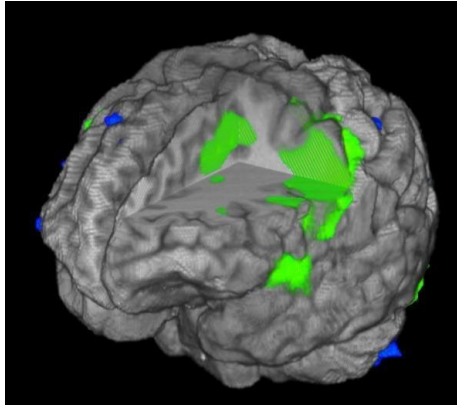
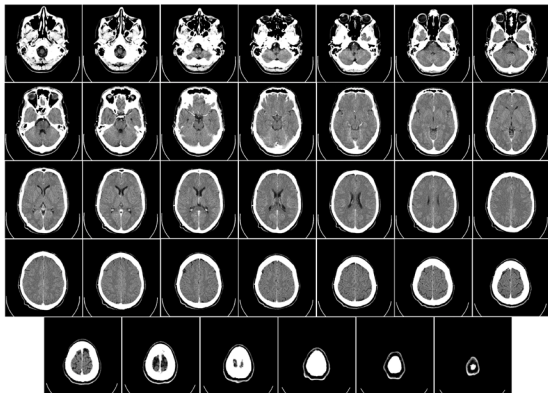


Image Credits: Scannet

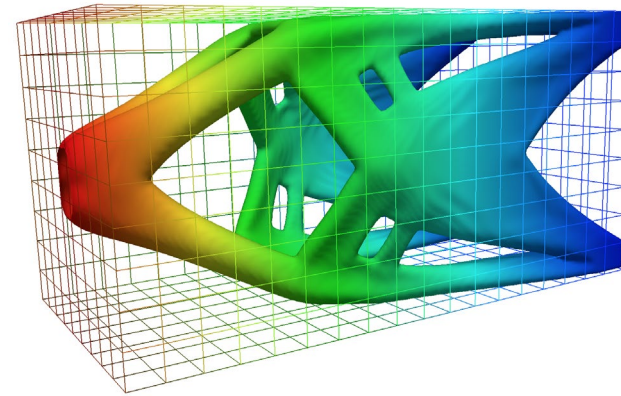
Volumetric Representation



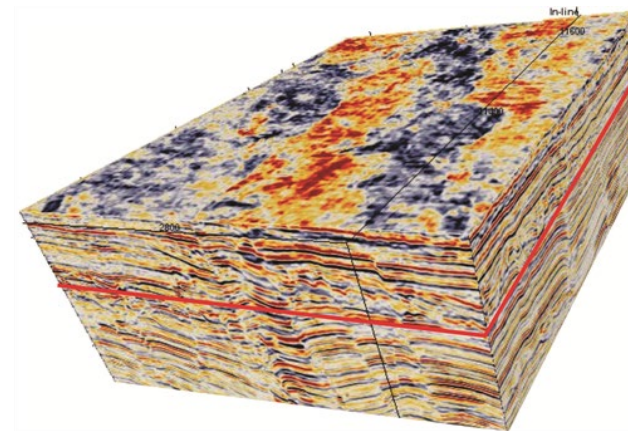
fMRI



CT

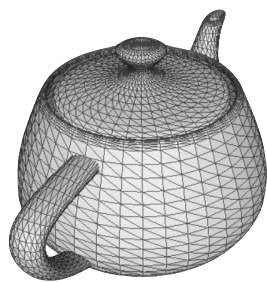
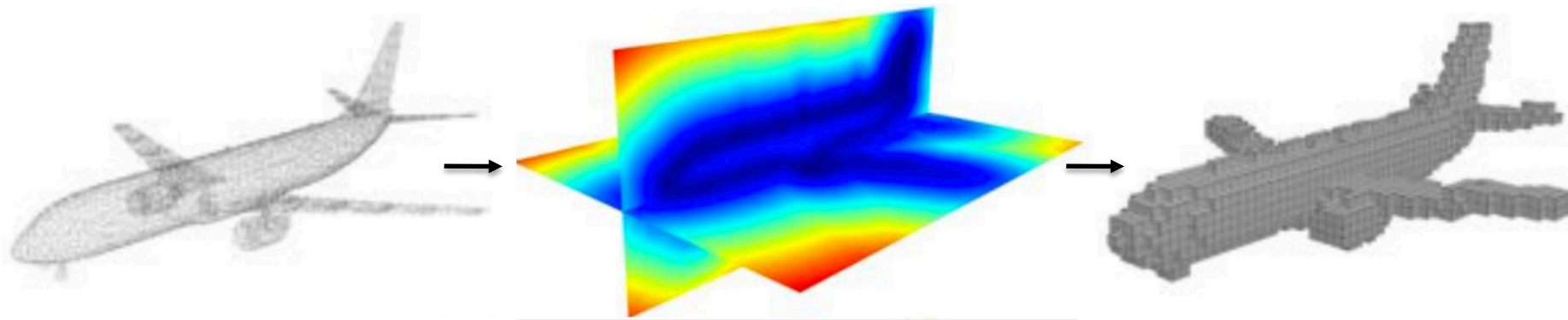


Manufacturing
(finite-element analysis)

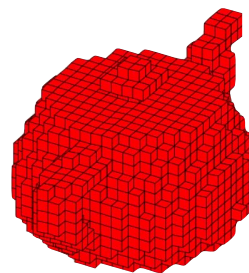


Geology

Volumetric Representation



Polygon Mesh



Occupancy Grid
30x30x30

Volumetric Representation

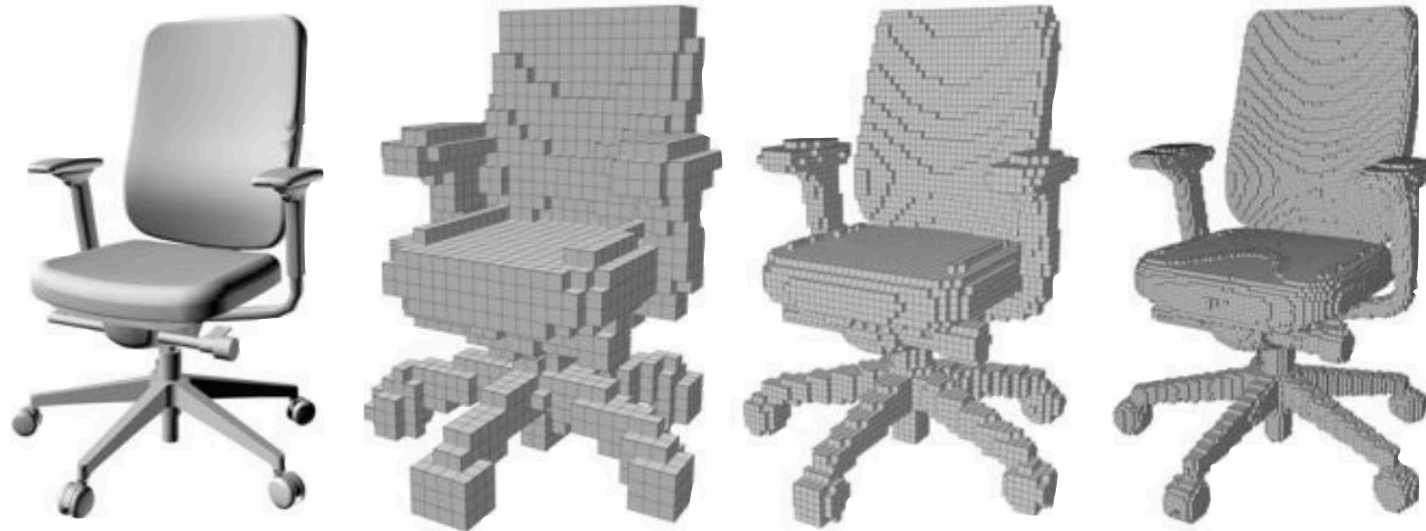
◆ Pros

- ◆ Regular grids representation
- ◆ Intuitive extension of images
- ◆ Easy to input to Neural Nets
- ◆ Each grid can have many feature inputs

◆ Cons

- ◆ Need to convert from point clouds scans
- ◆ Surface voxels? / solid voxels?
- ◆ Space / Time Complexities

Volumetric Representation



$\frac{\#occupied\ grid}{\#total\ grid}$

Occupancy:

10.41%

5.09%

2.41%

Resolution:

32

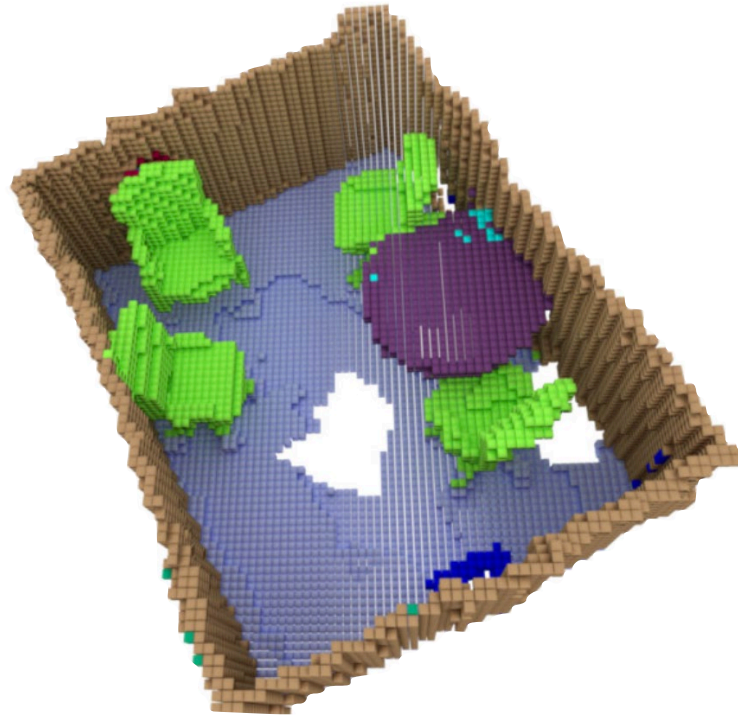
64

128

Resolution N

$O(N^3)$

Volumetric Representation

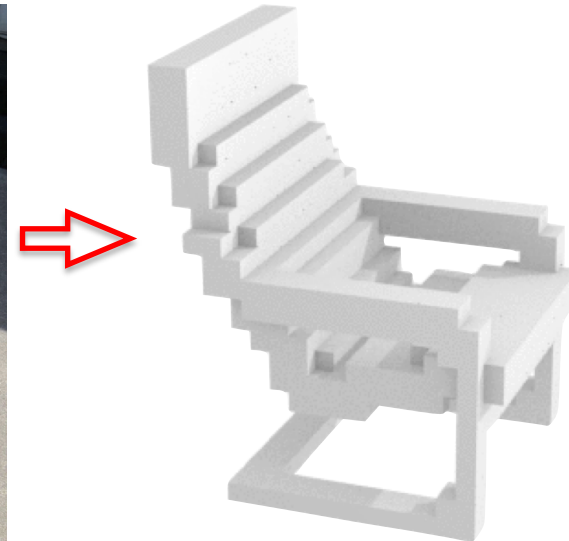
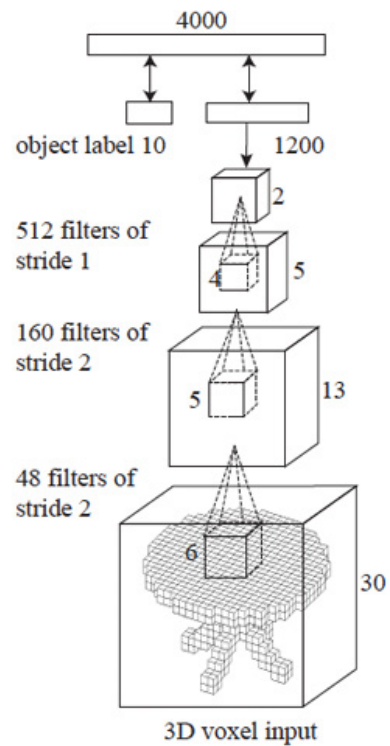


< 10%

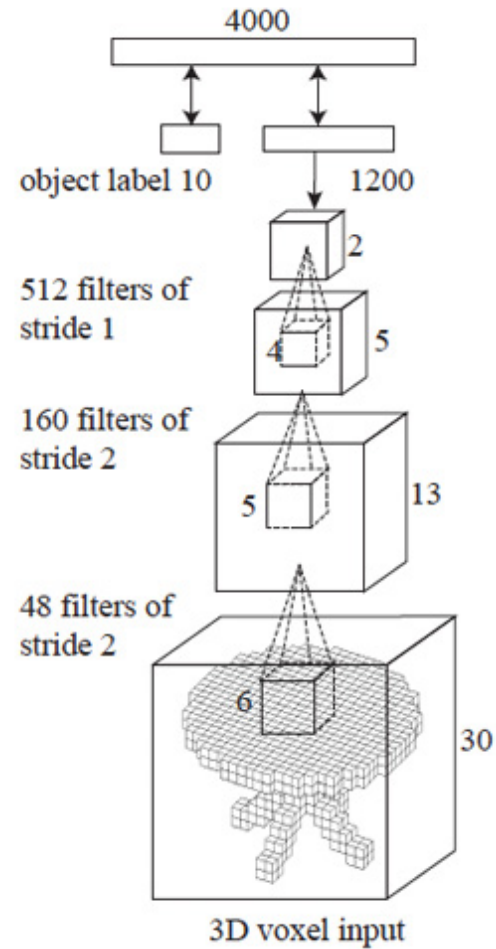
Image Credits: Scannet

Multi-view Representation

- ◆ Classification
- ◆ Reconstruction

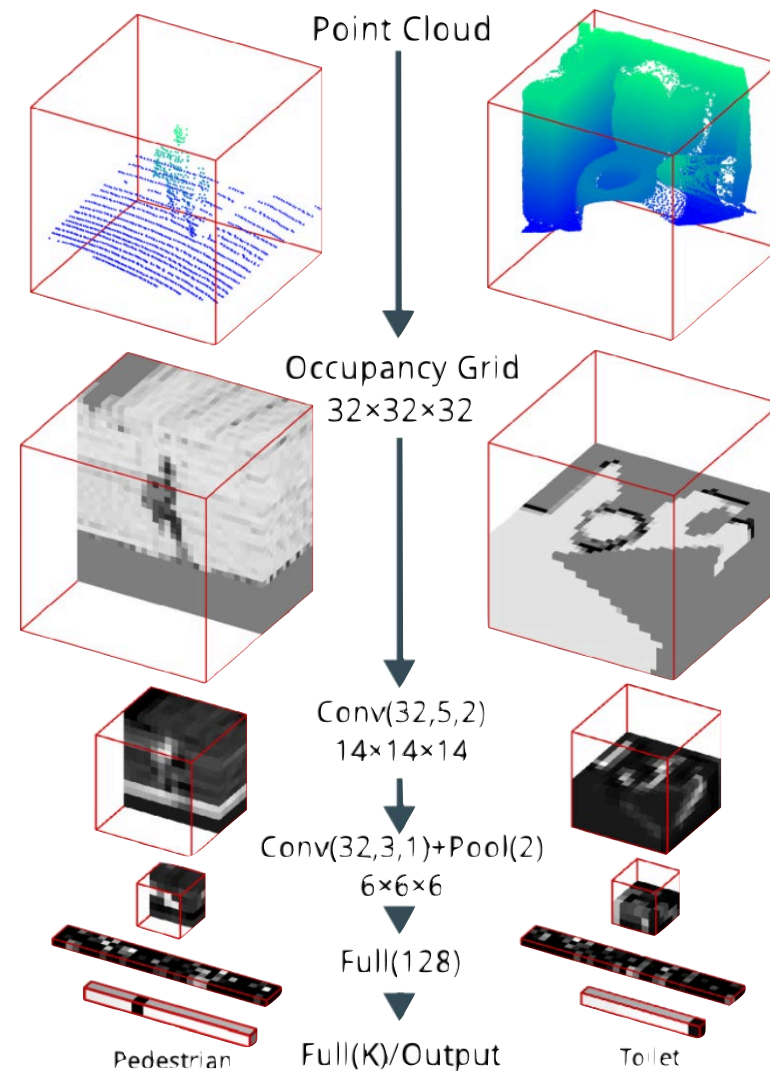


Volumetric Representation: Classification



Z. Wu, S. Song, A.
Khosla, F. Yu, L. Zhang,
X. Tang and J. Xiao,
**“3D ShapeNets: A
Deep Representation
for Volumetric Shape
Modeling”**,
CVPR2015

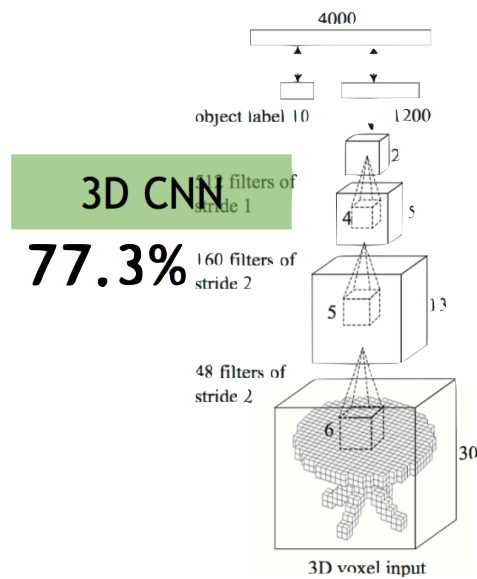
Volumetric Representation: Classification



Daniel Maturana and
Sebastian Scherer,
“VoxNet: A 3D
Convolutional Neural
Network for Real-
Time Object
Recognition”,
IROS2015

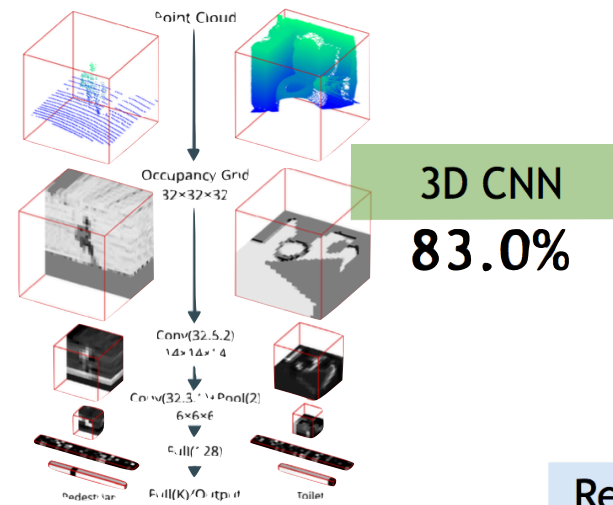
Volumetric Representation: Classification

3DShapeNets from Princeton
CVPR 2015 Oral

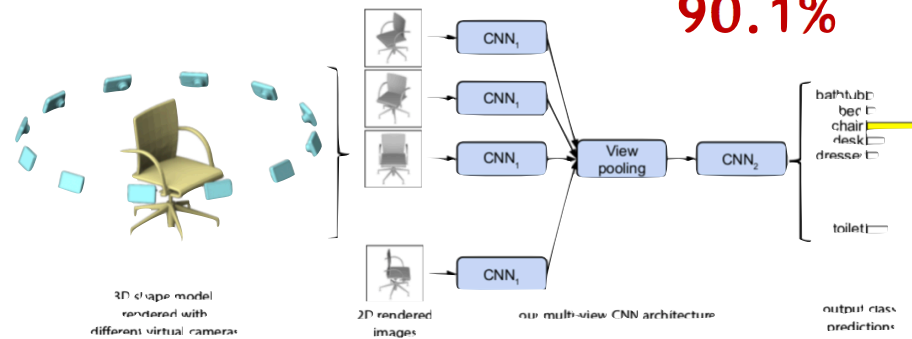


MVCNN from UMass
ICCV 2015

VoxNet from CMU Robotics
IEEE/RSJ 2015



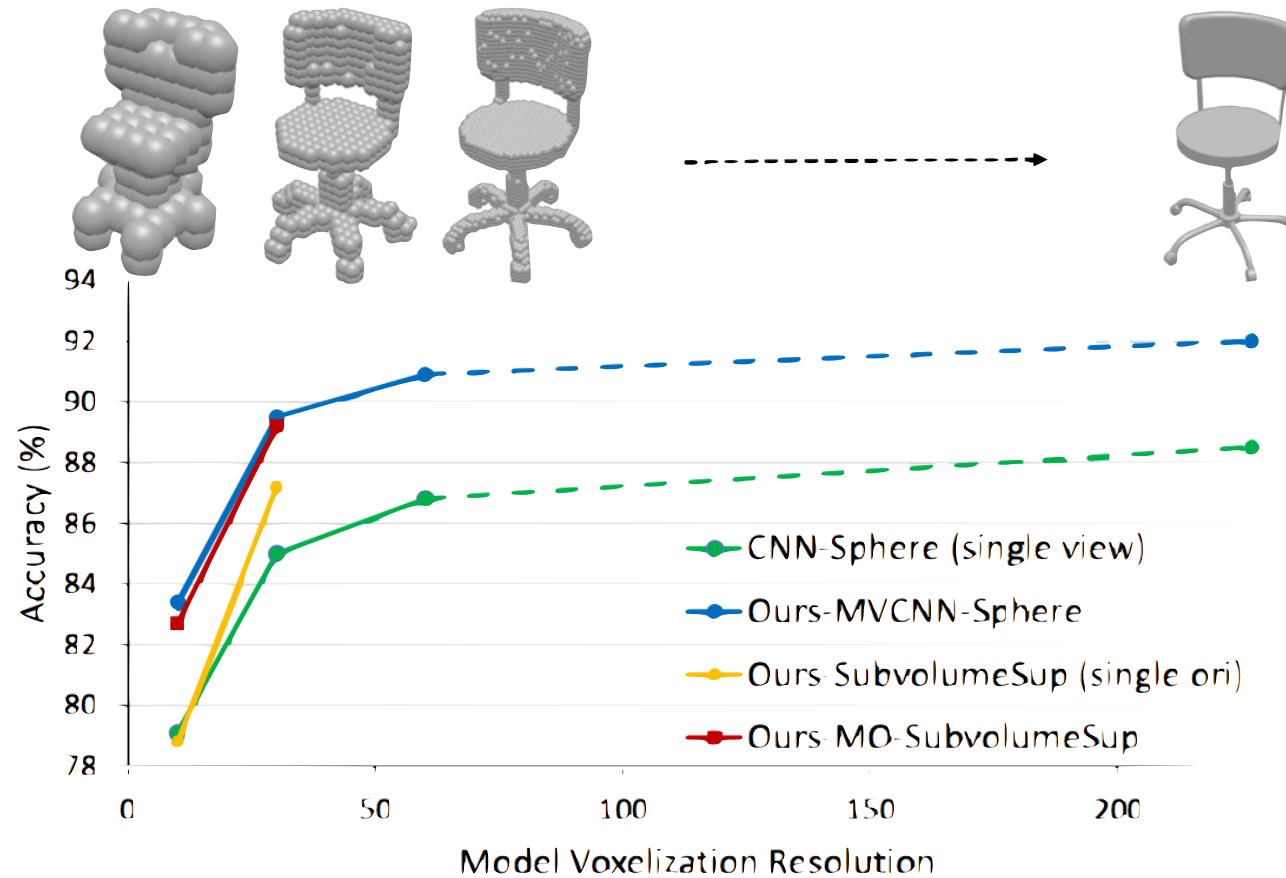
Rendering +
2D CNN
90.1%



Volumetric Representation: Classification

- ◆ Vanilla 3D CNN methods perform worse than Multi-view CNNs
 - ◆ Resolution is limited (information loss)
 - ◆ Computation is more expensive than 2D
 - ◆ Have no colors / shading effects
 - ◆ Aliasing

Volumetric Representation: Classification



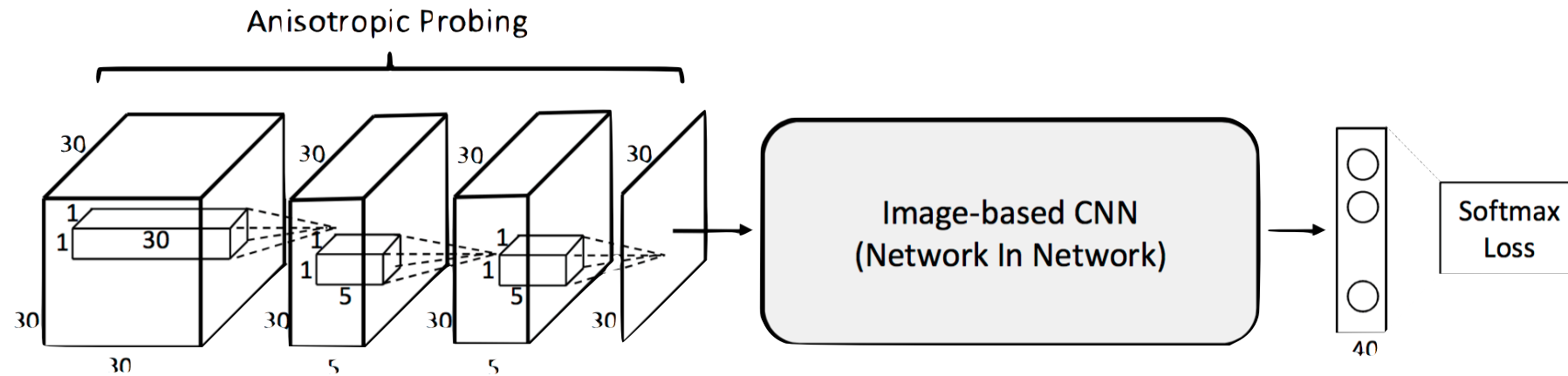
Volumetric and Multi-View CNNs for Object Classification on 3D Data

Hao Su*, **Charles Qi***, **Matthias Niessner**, **Angela Dai**, **Mengyuan Yan**, **Leonidas Guibas**

CVPR 2016 (spotlight oral)

Volumetric Representation: Classification

*Idea: "X-ray" rendering + Image (2D) CNNs
very low #param, very low computation*



Volumetric and Multi-View CNNs for Object Classification on 3D Data

Hao Su*, **Charles Qi***, **Matthias Niessner**, **Angela Dai**, **Mengyuan Yan**, **Leonidas Guibas**

CVPR 2016 (spotlight oral)

Volumetric Representation: Classification

Network	Single-Ori	Multi-Ori
E2E-[33]	83.0	87.8
VoxNet[24]	83.8	85.9
3D-NIN	86.1	88.5
Ours-SubvolumeSup	87.2	89.2
Ours-AniProbing	85.9	89.9

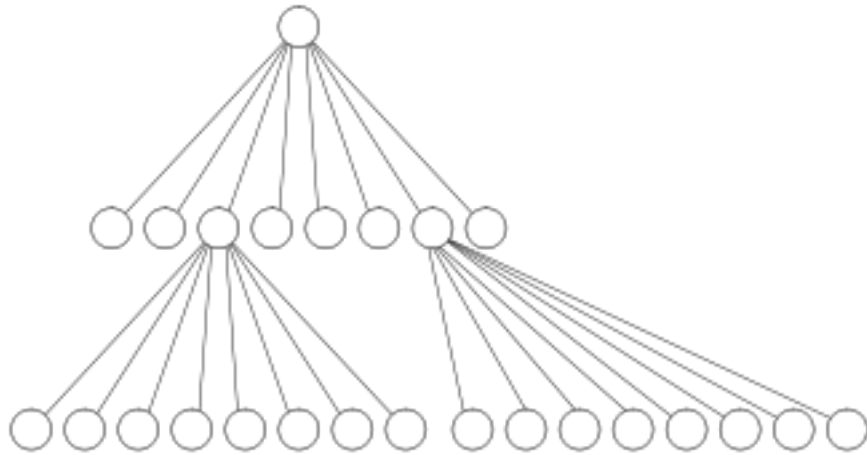
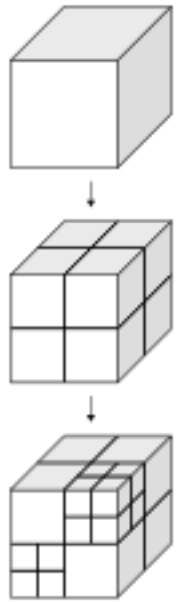
Multi-view: 90.1%

Volumetric and Multi-View CNNs for Object Classification on 3D Data

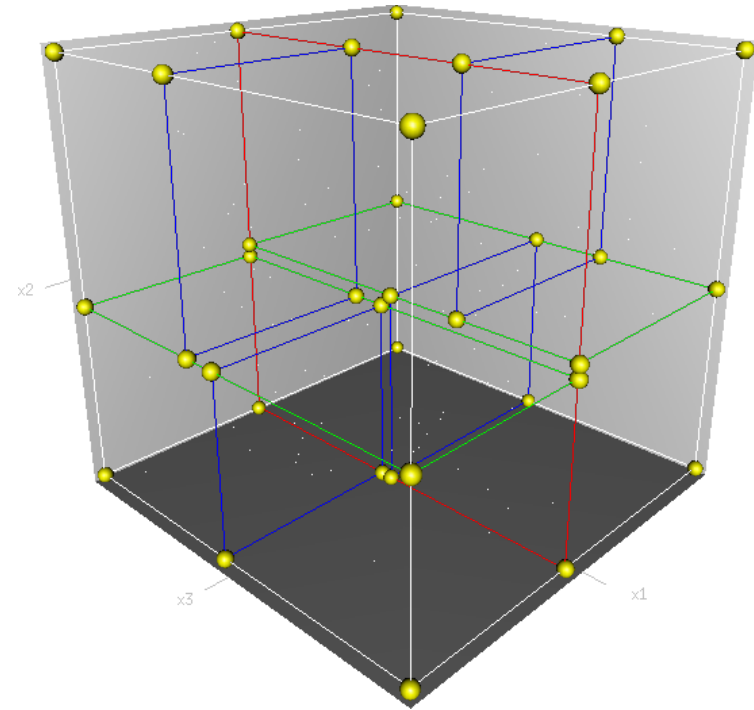
Hao Su*, Charles Qi*, Matthias Niessner, Angela Dai, Mengyuan Yan, Leonidas Guibas

CVPR 2016 (spotlight oral)

Sparse Volumetric Representation



Octree

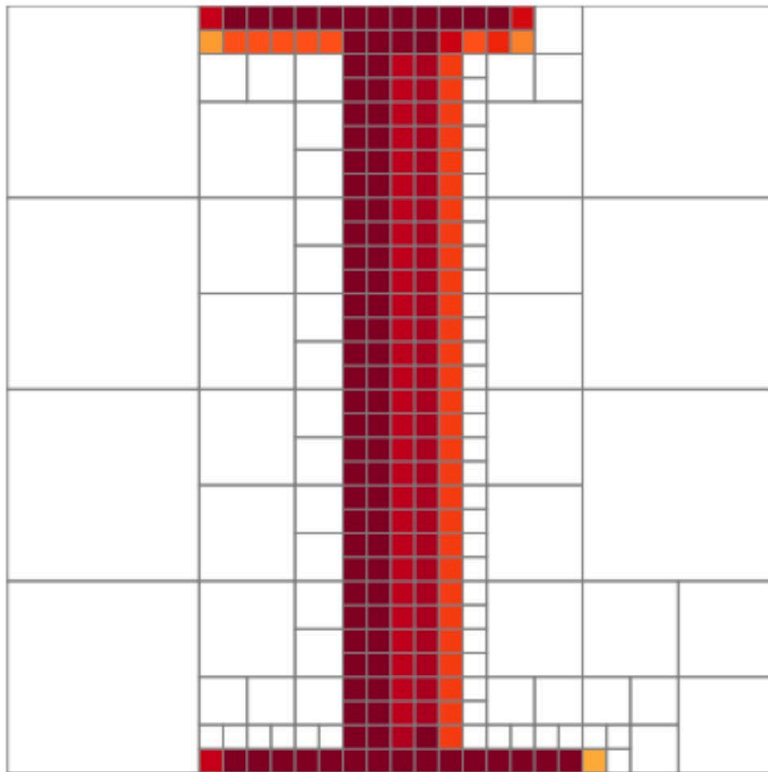


3D KD-Tree

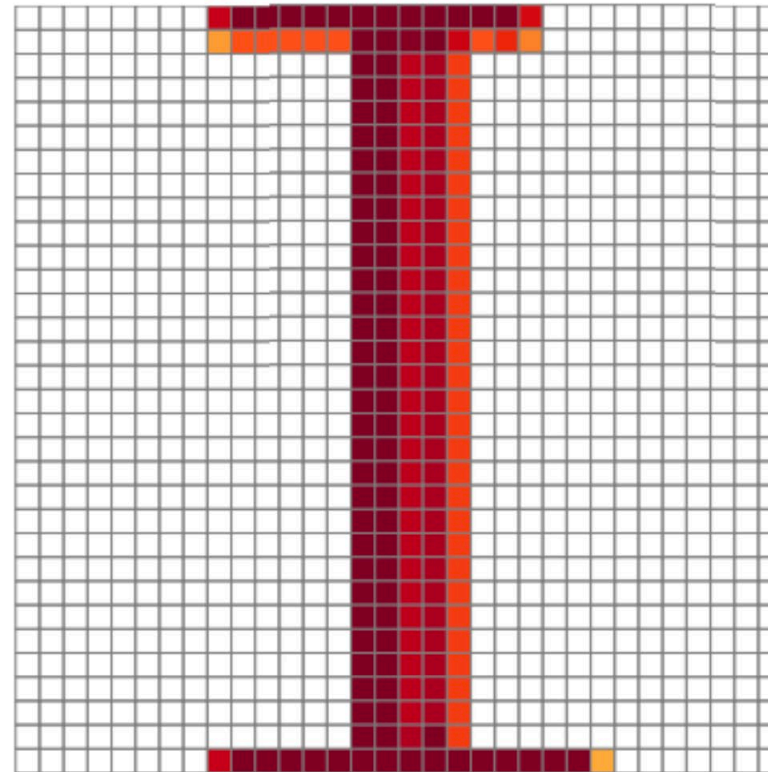
Image Credits: Wikipedia

Sparse Volumetric Representation

OctNet

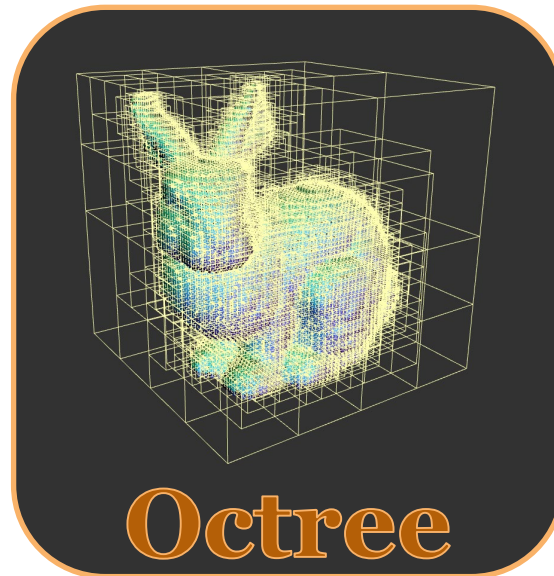
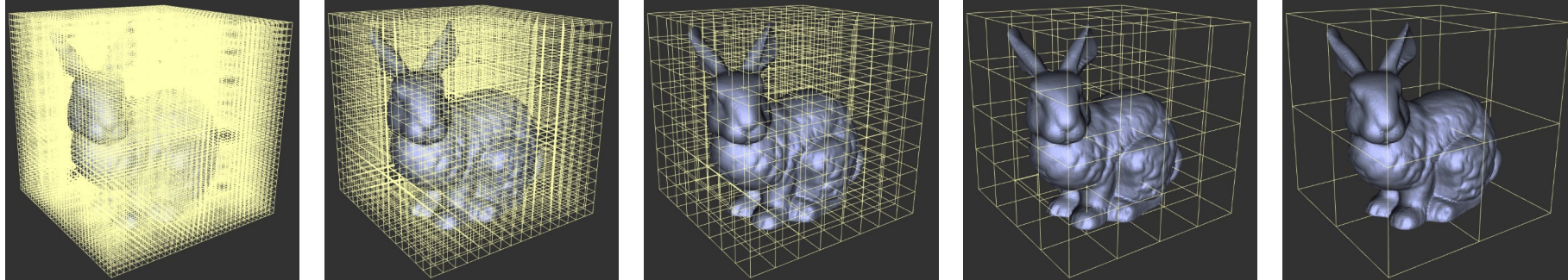


Dense 3D ConvNet



OctNet: Learning Deep 3D Representations at High Resolutions

Sparse Volumetric Representation



Sparse Volumetric Representation

O-CNN: Octree-based Convolutional Neural Networks for 3D Shape Analysis

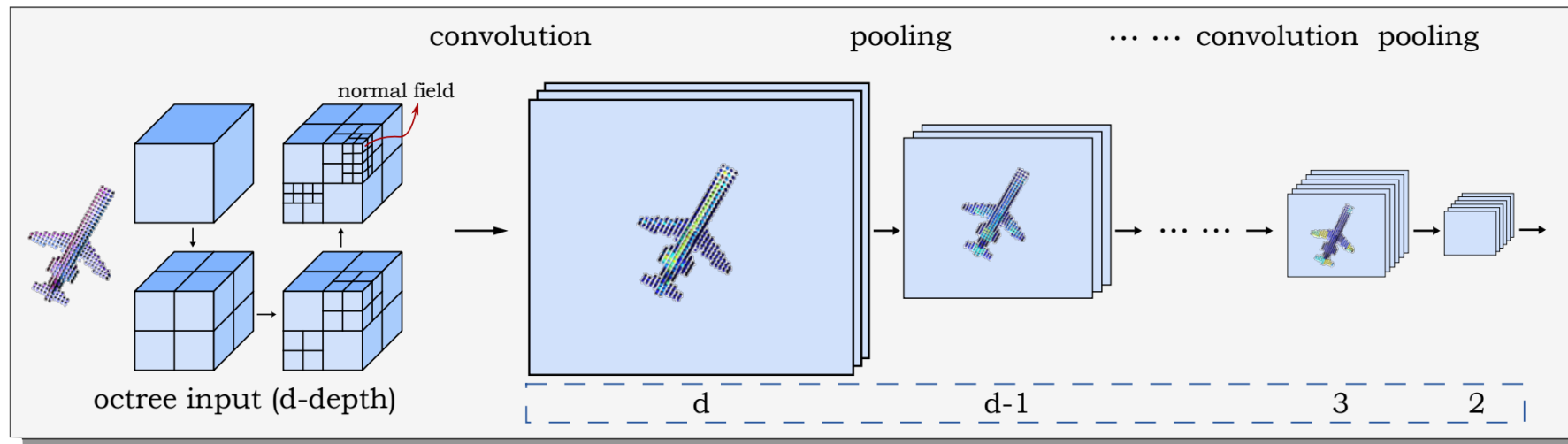
PENG-SHUAI WANG, Tsinghua University and Microsoft Research Asia

YANG LIU, Microsoft Research Asia

YU-XIAO GUO, University of Electronic Science and Technology of China and Microsoft Research Asia

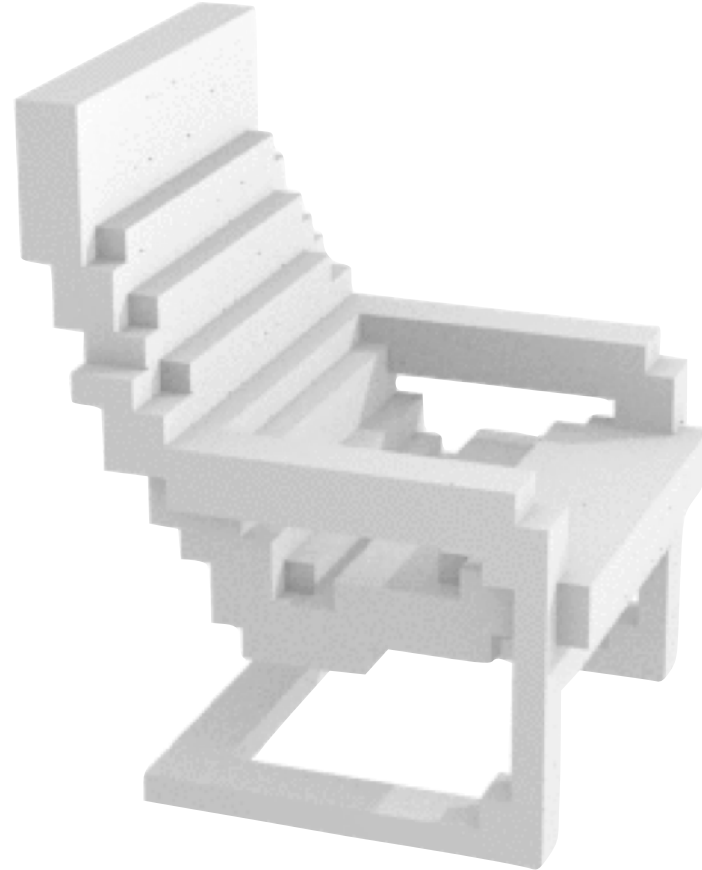
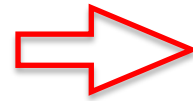
CHUN-YU SUN, Tsinghua University and Microsoft Research Asia

XIN TONG, Microsoft Research Asia



Classification Accuracy: 89.9%

Volumetric Representation: Reconstruction



Volumetric Representation: Completion

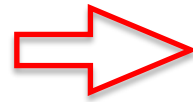
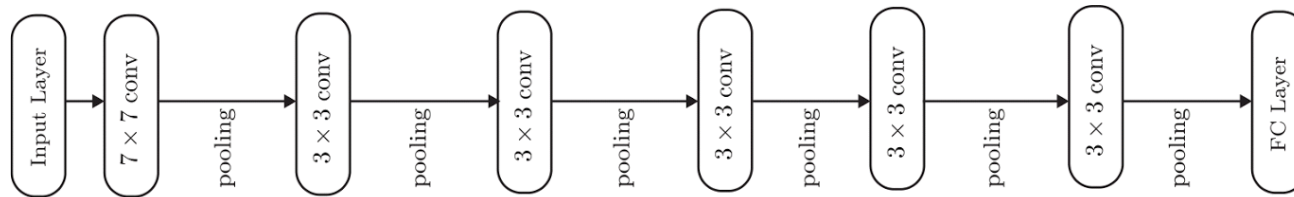
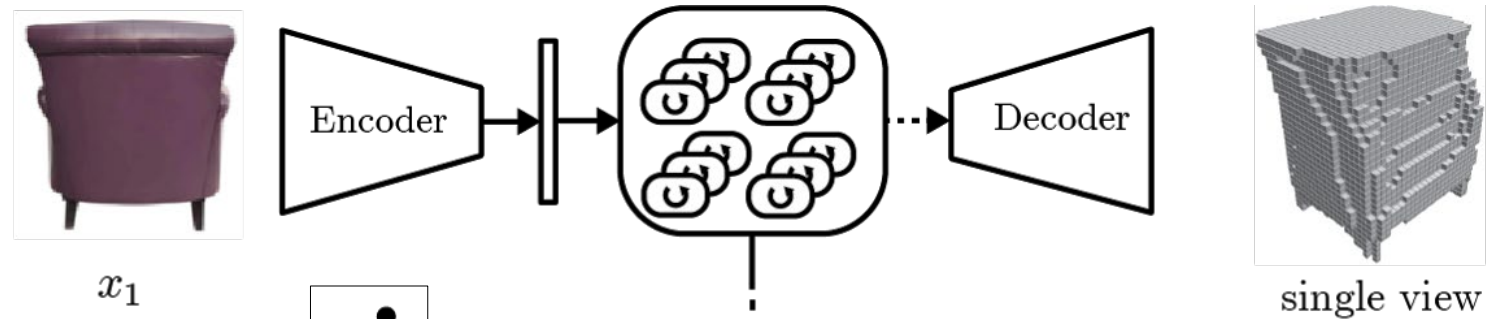


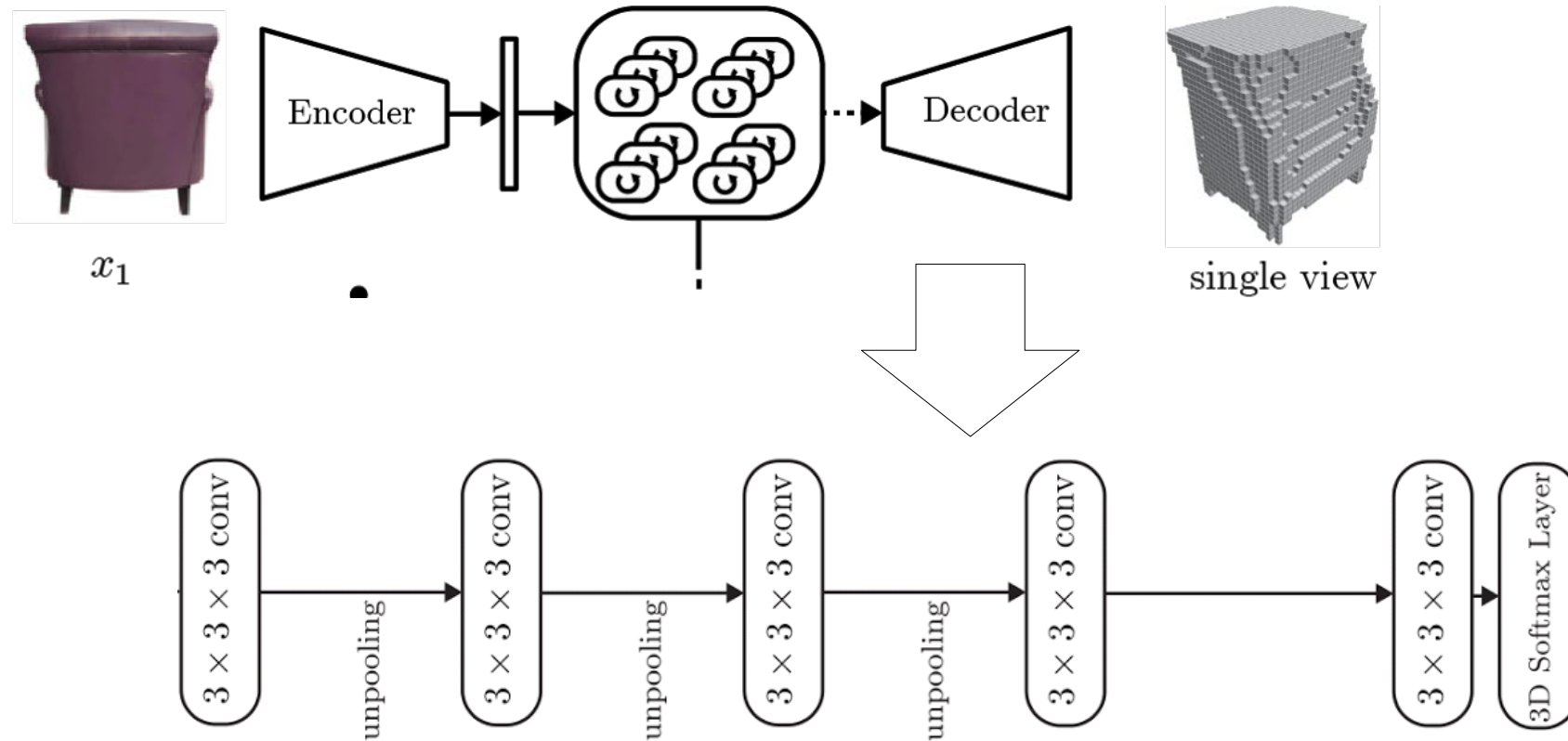
Image Credits: Angela Dai et. al.

Volumetric Representation: Reconstruction



3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction
Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese
ECCV 2016

Volumetric Representation: Reconstruction

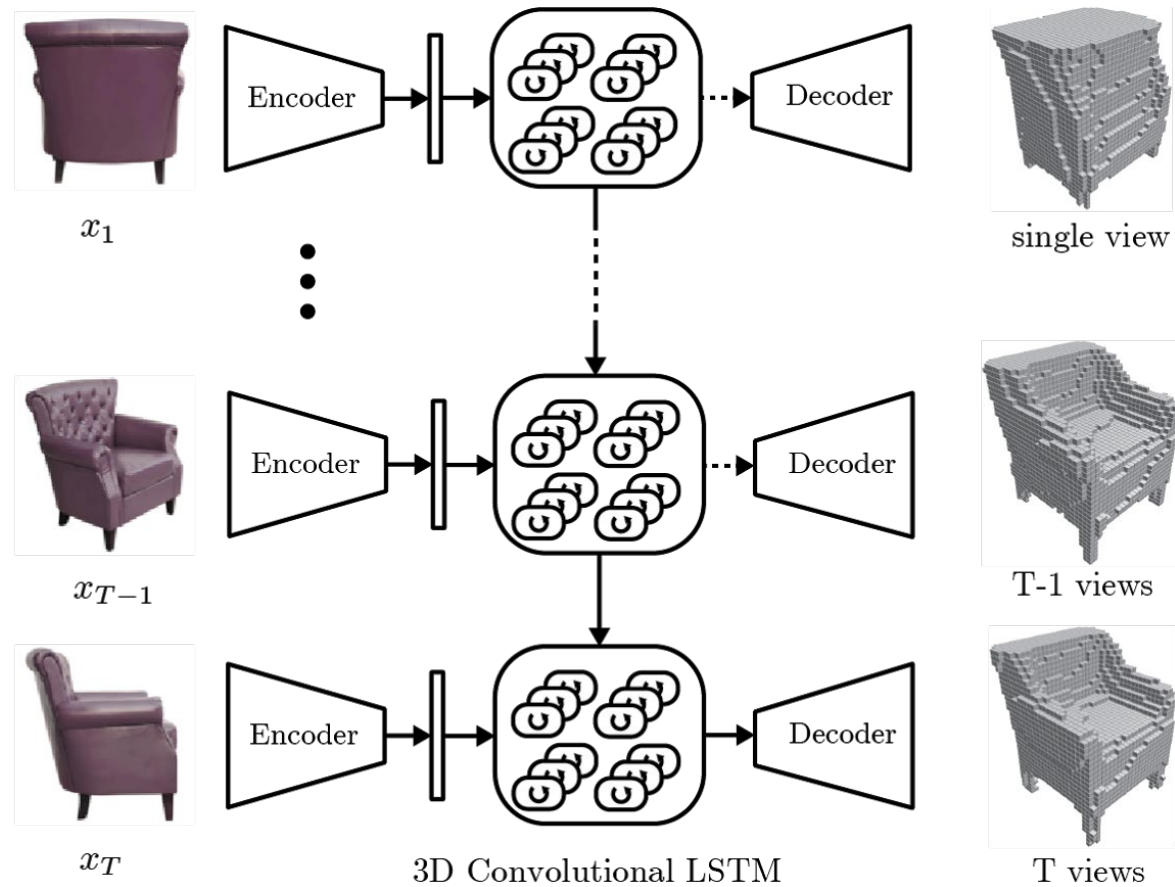


3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction

Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese

ECCV 2016

Volumetric Representation: Reconstruction

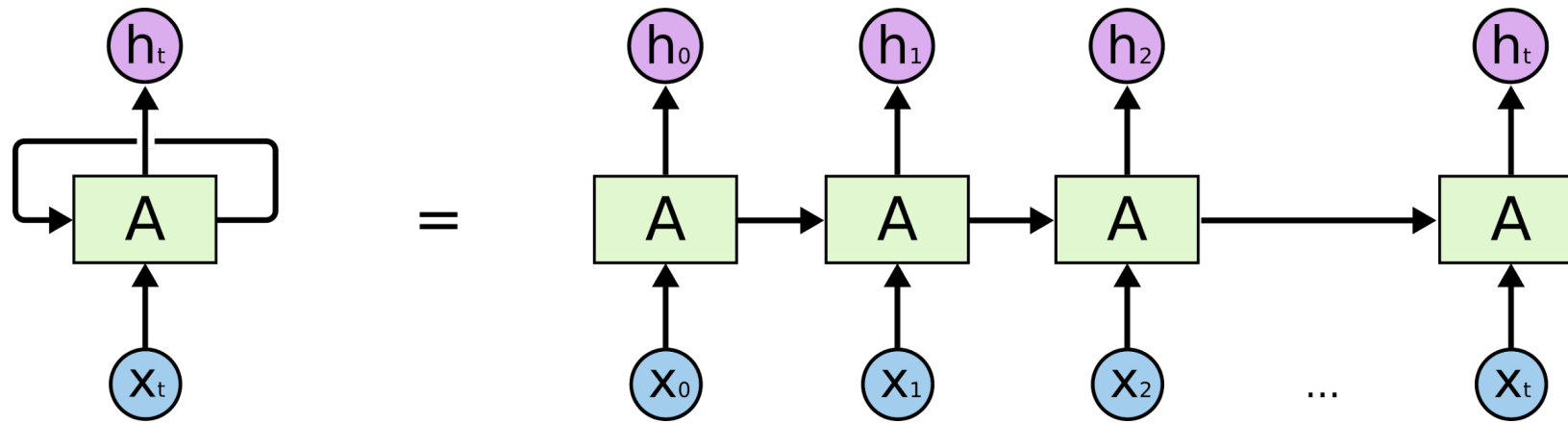


3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction

Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese

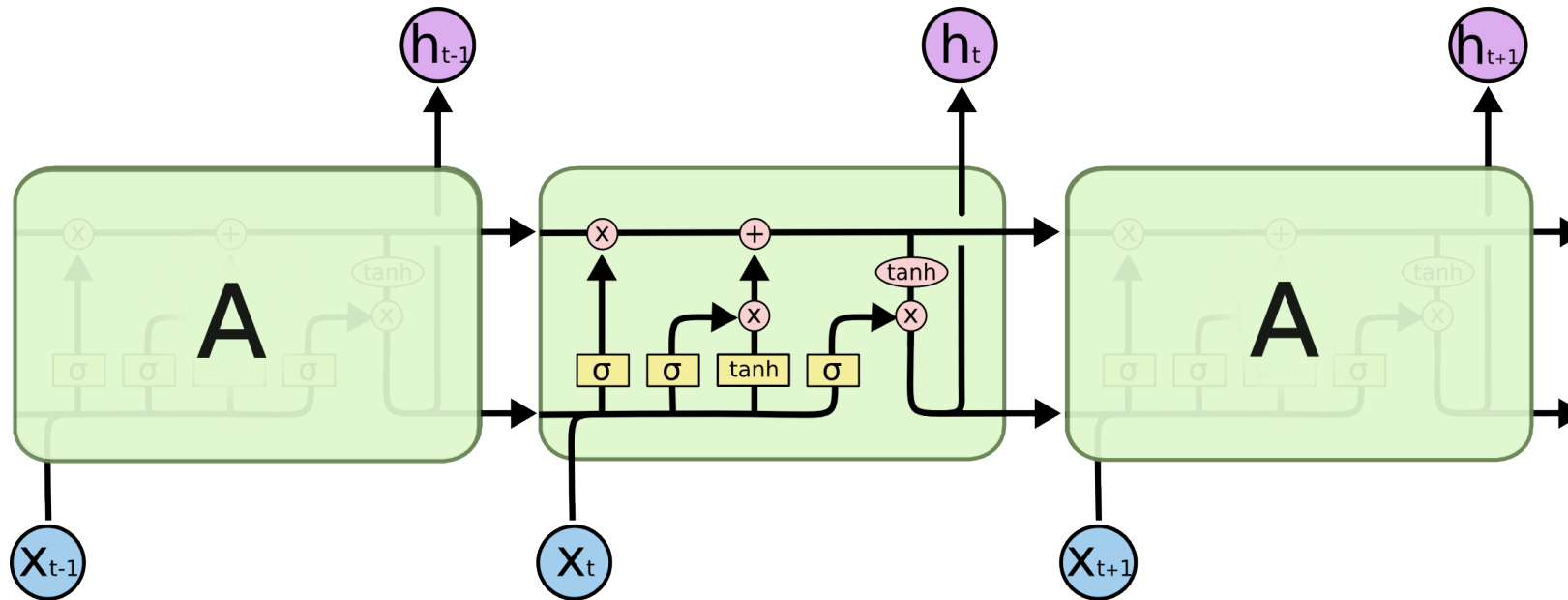
ECCV 2016

RNN



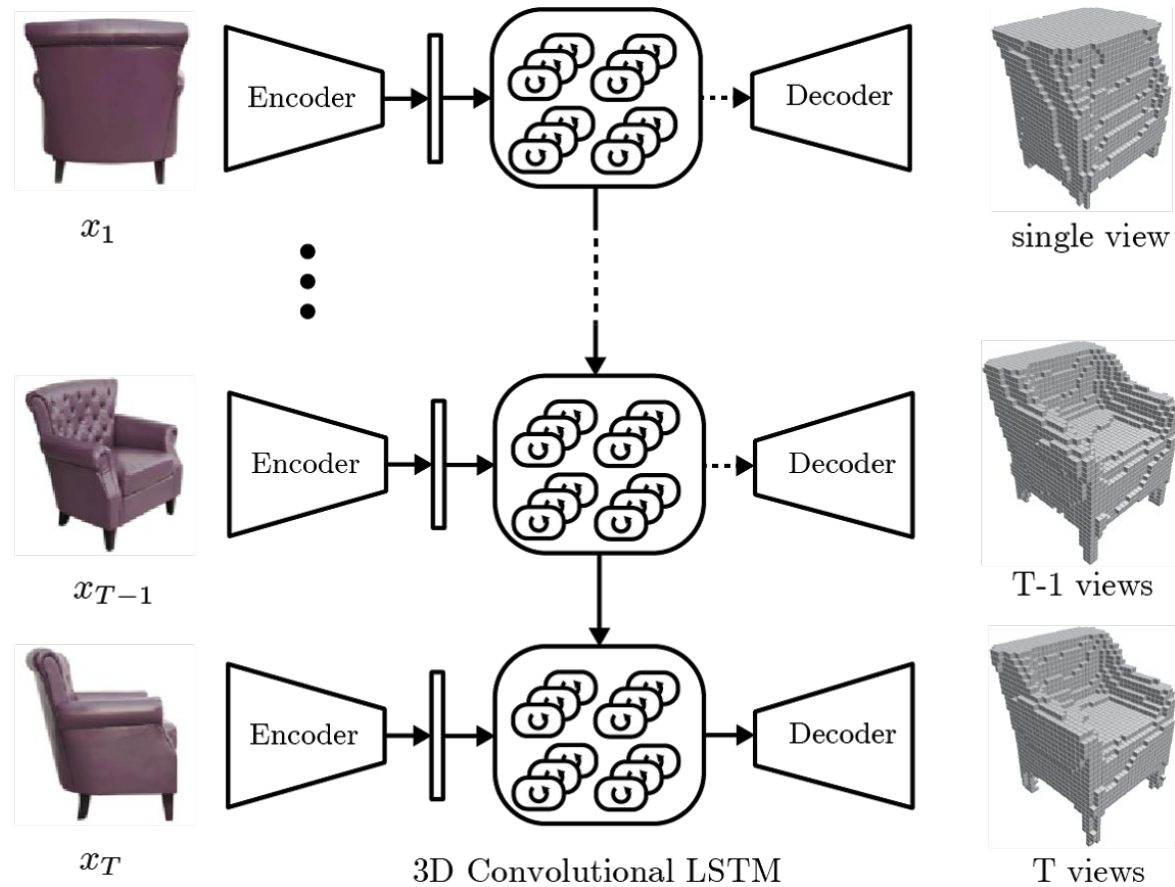
[Christopher Olah] Understanding LSTM Networks,
<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

LSTM



[Christopher Olah] Understanding LSTM Networks,
<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Volumetric Representation: Reconstruction



3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction

Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese

ECCV 2016

Volumetric Representation: Reconstruction

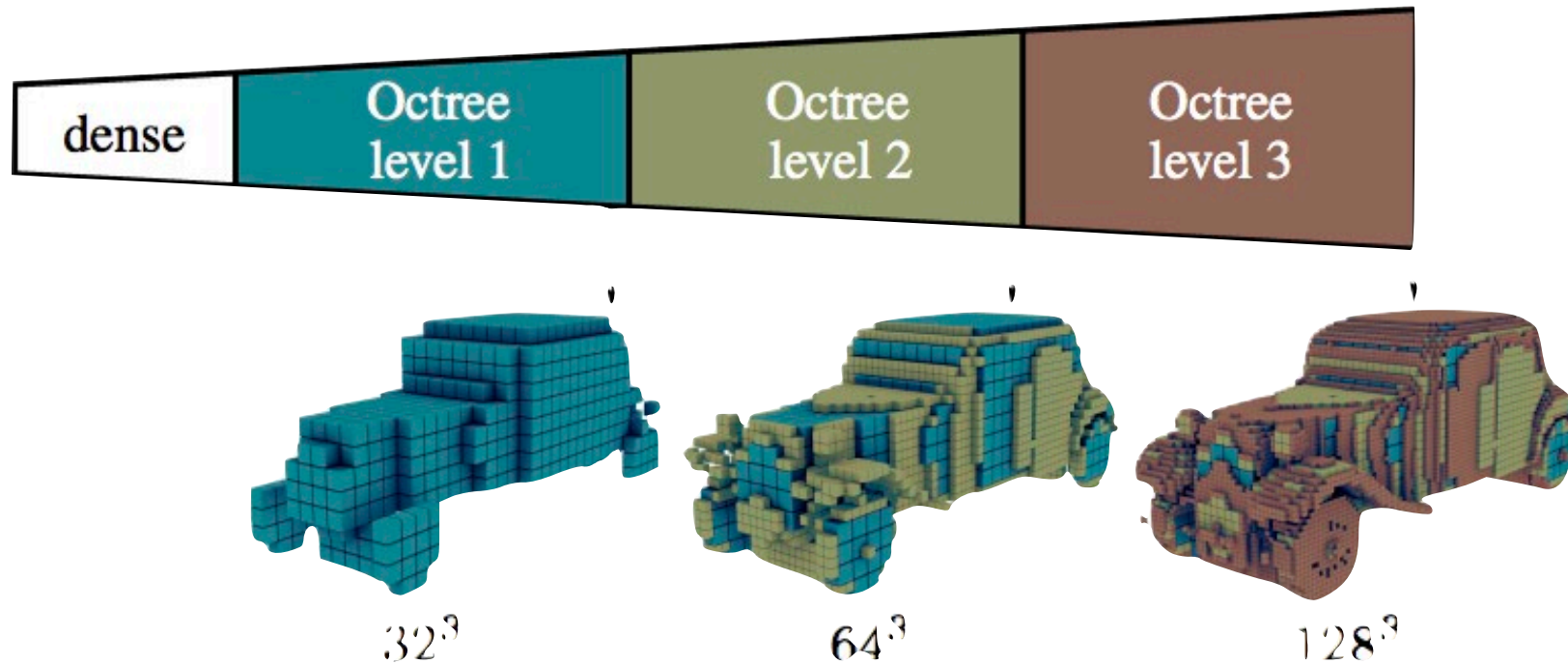


3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction

Christopher B. Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, Silvio Savarese

ECCV 2016

Volumetric Representation: Reconstruction

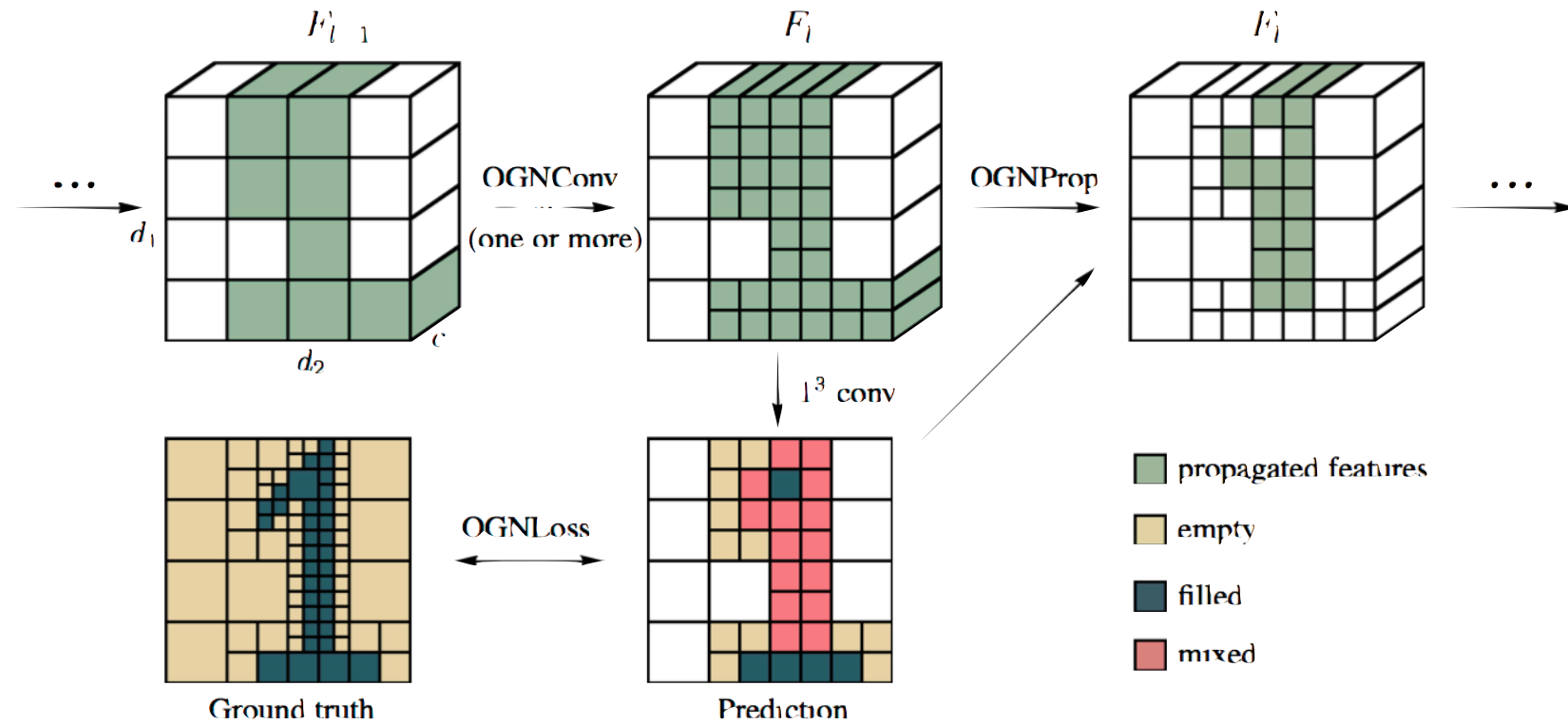


Maxim Tatarchenko, Alexey Dosovitskiy, Thomas Brox

“Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs”

ICCV, 2017

Volumetric Representation: Reconstruction

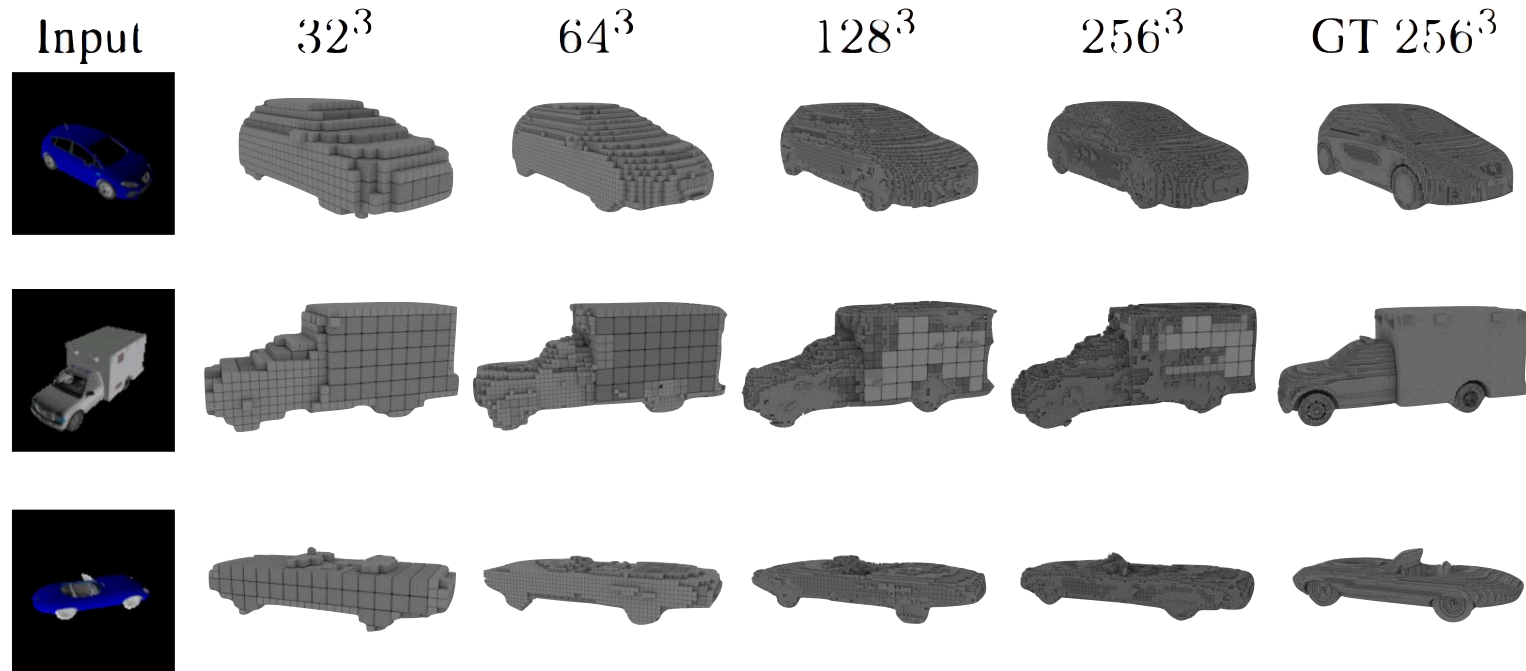


Maxim Tatarchenko, Alexey Dosovitskiy, Thomas Brox

“Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs”

ICCV, 2017

Volumetric Representation: Reconstruction

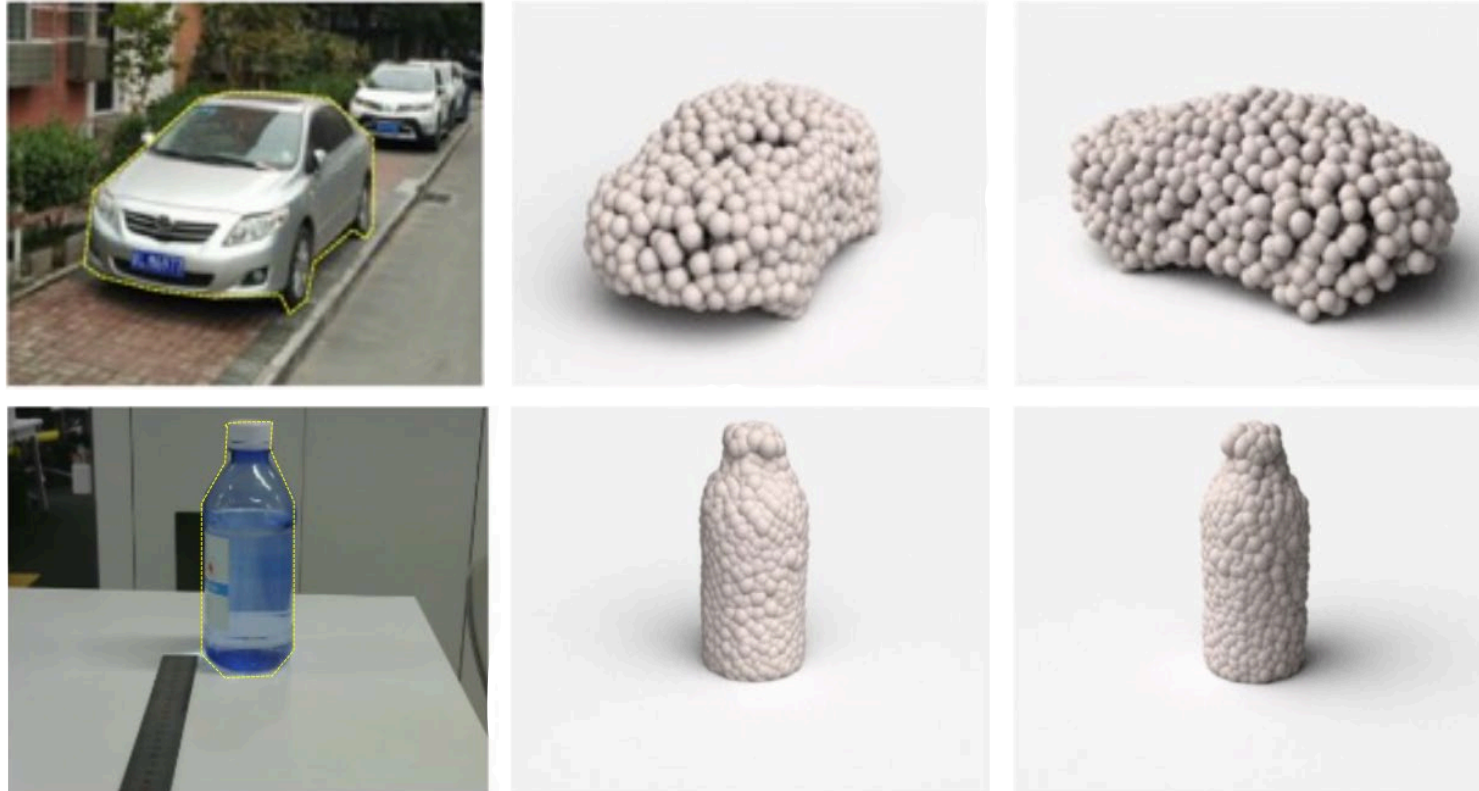


Maxim Tatarchenko, Alexey Dosovitskiy, Thomas Brox

“Octree Generating Networks: Efficient Convolutional Architectures for High-resolution 3D Outputs”

ICCV, 2017

Reconstruction: Point Cloud Based



Input

Reconstructed 3D point cloud

Haoqiang Fan, Hao Su, Leonidas Guibas,

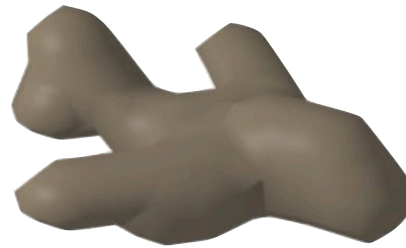
“A Point Set Generation Network for 3D Object Reconstruction from a Single Image”

CVPR, 2017

Reconstruction: Surface Based



(a) Input Image



(b) 16^3



(c) 32^3



(d) 64^3



(e) 128^3



(f) 256^3

Christian Hane, Shubham Tulsiani, Jitendra Malik

“Hierarchical Surface Prediction for 3D Object Reconstruction”

3DV, 2017

Conclusion

- ◆ 3D Deep Learning is a new and active research direction
- ◆ 3D data have different representation
- ◆ Multi-view CNNs take advantage of the state-of-the-art 2D CNNs
- ◆ Volumetric 3D CNNs suffer from the “curse of dimensionality”



The End