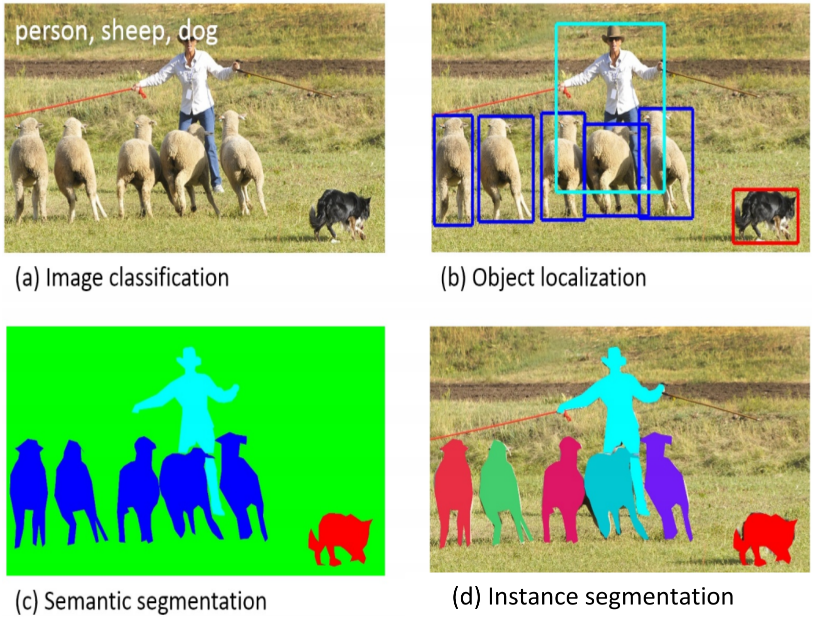
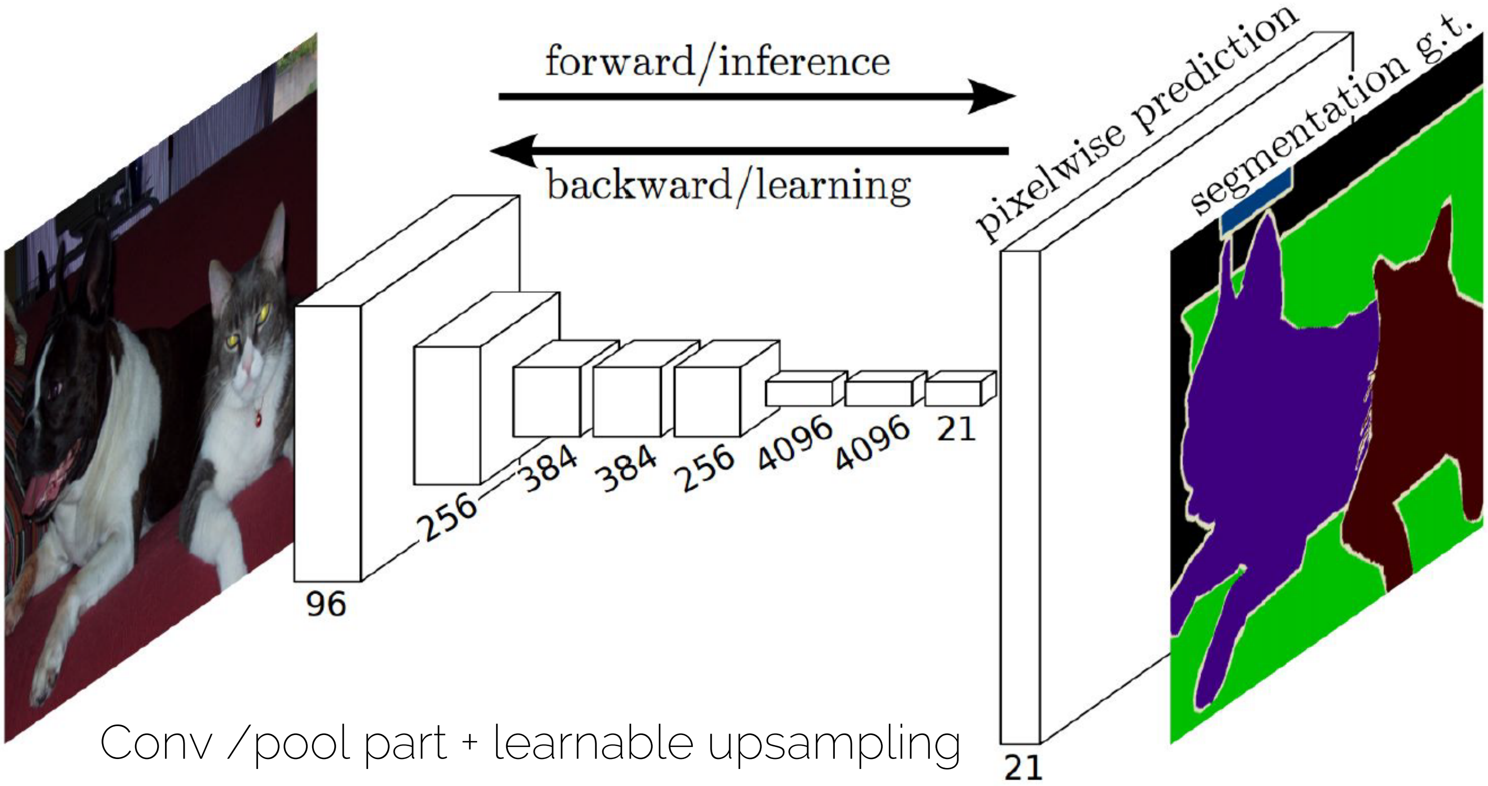
Lecture 9 Recap

Segmentation Overview

- Semantic segmentation
  - Classify all pixels
  - Fully convolutional models, downsample, then upsample
  - Learnable upsampling (deconvolution)
  - Skip connection can help (more later)
- Instance segmentation
  - Detect instance, generate mask
  - Similar pipelines to object detection



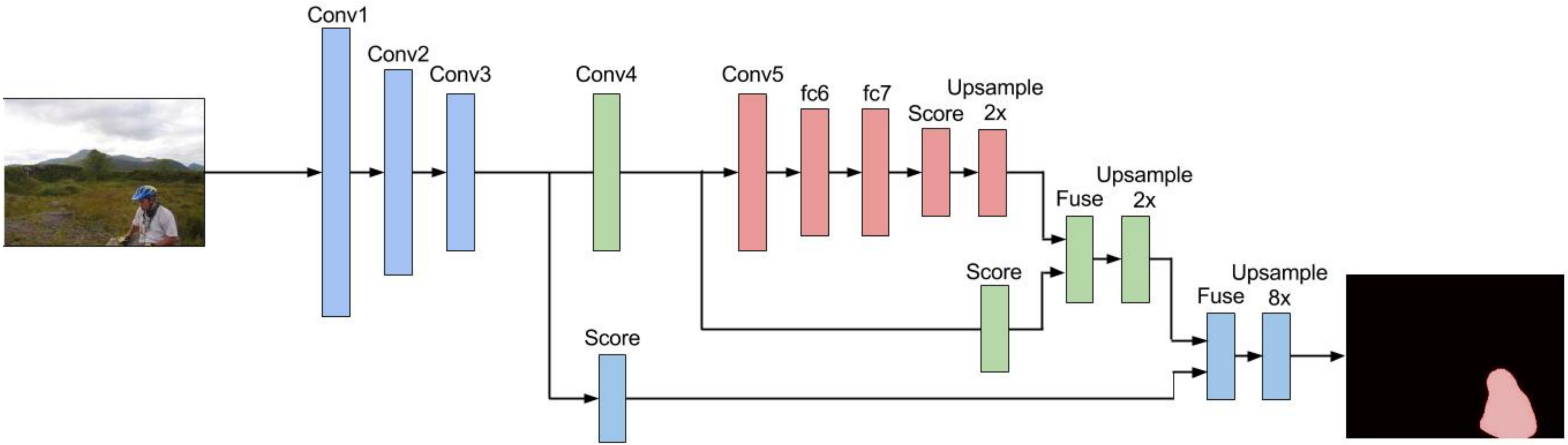
Semantic Segmentation (FCN)



Conv /pool part + learnable upsampling

[Long et al. 15] Fully Convolutional Networks for Semantic Segmetnation (FCN)

FCN: Architecture

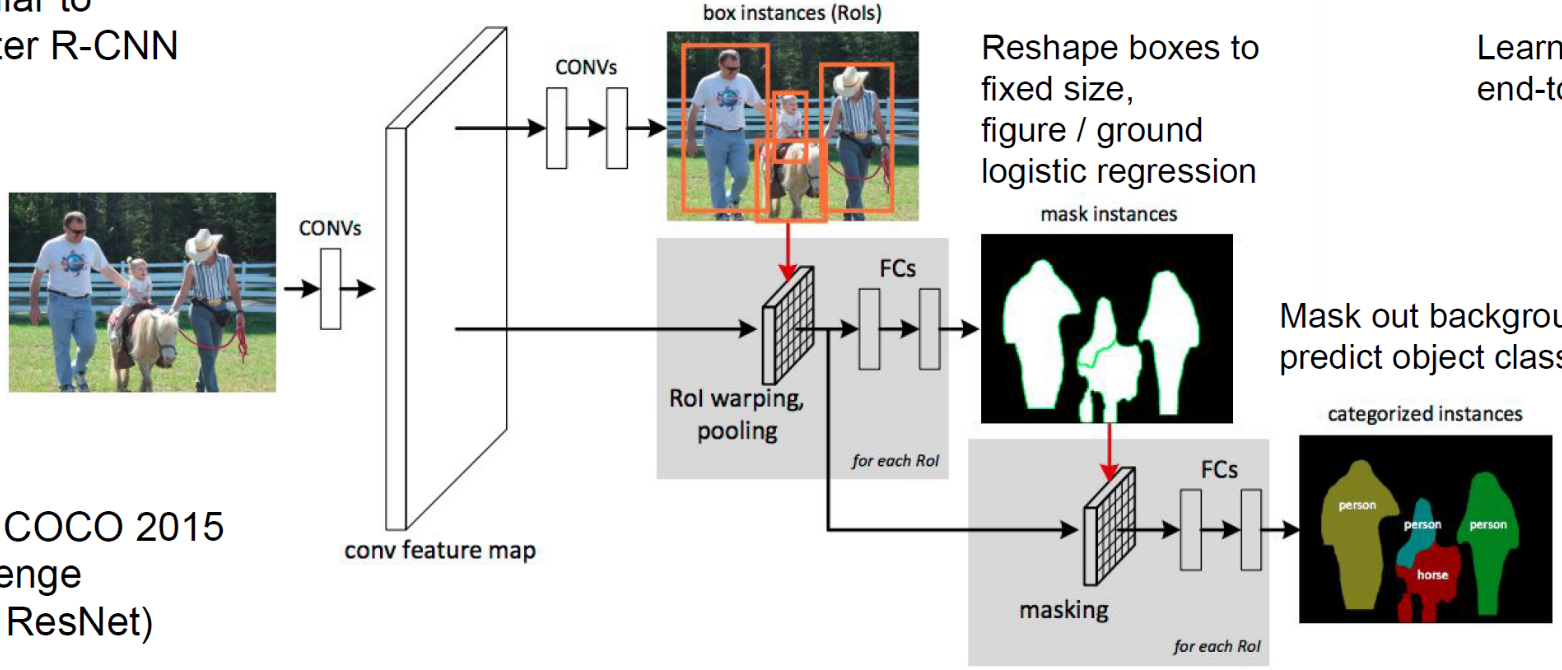


[Long et al. 15] Fully Convolutional Networks for Semantic Segmentation (FCN)

Instance Segmentation: Cascades

Similar to
Faster R-CNN

Region proposal network (RPN)



Reshape boxes to
fixed size,
figure / ground
logistic regression

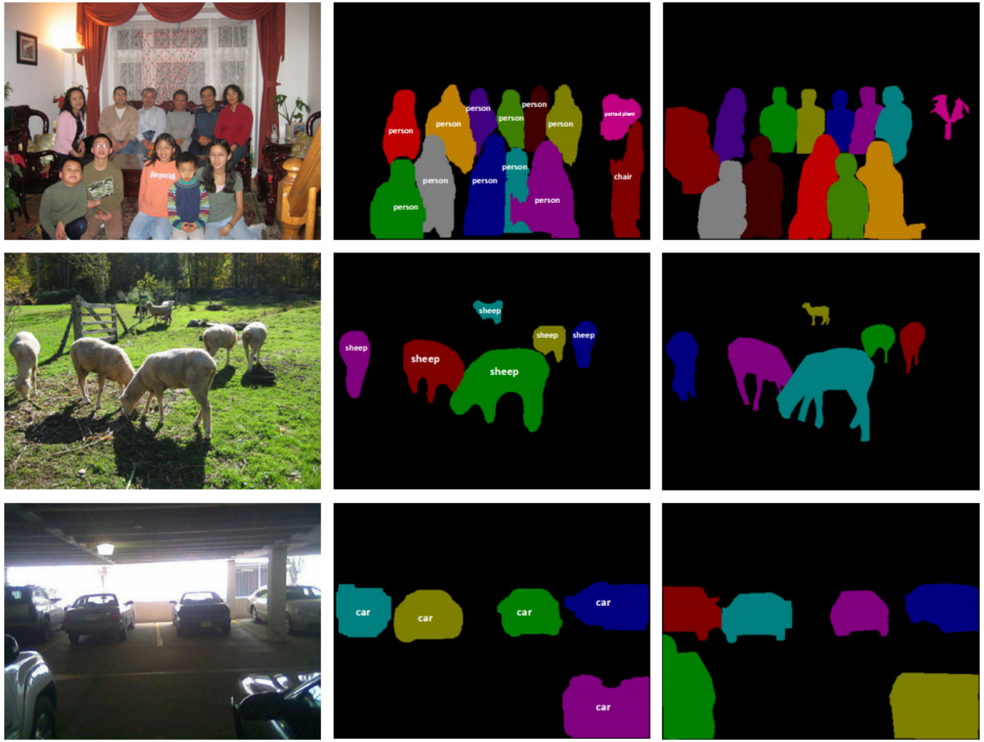
Learn entire model
end-to-end!

Mask out background,
predict object class

Won COCO 2015
challenge
(with ResNet)

[Dai et al. 15] Instance-aware Semantic Segmentation via Multi-task Network Cascades (Slide by Li/Karpathy/Johnson)

Instance Segmentation: Cascades



Input Prediction Ground Truth

[Dai et al. 15] Instance-aware Semantic Segmentation via Multi-task Network Cascades (Slide by Li/Karpathy/Johnson)

Using CNNs in Computer Vision

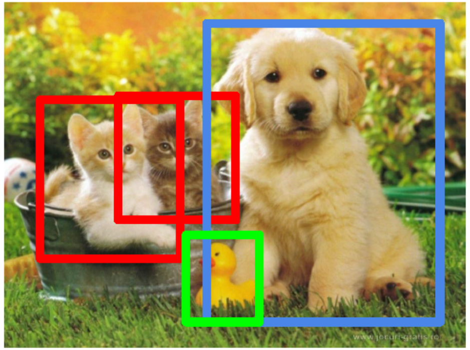
Classification

CIFAR 10 +
"raw" CNN ☺

Classification + Localization

Regression and/or
sliding window

Object Detection



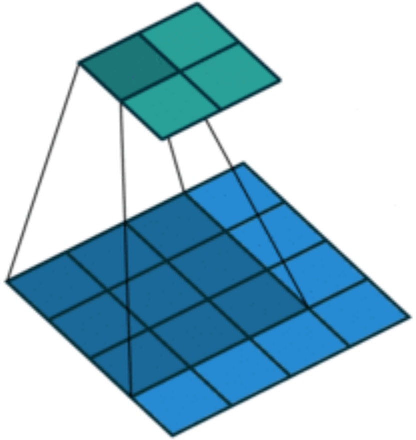
Selective Search, RP
(Fast(er)) R-CNN

Instance Segmentation

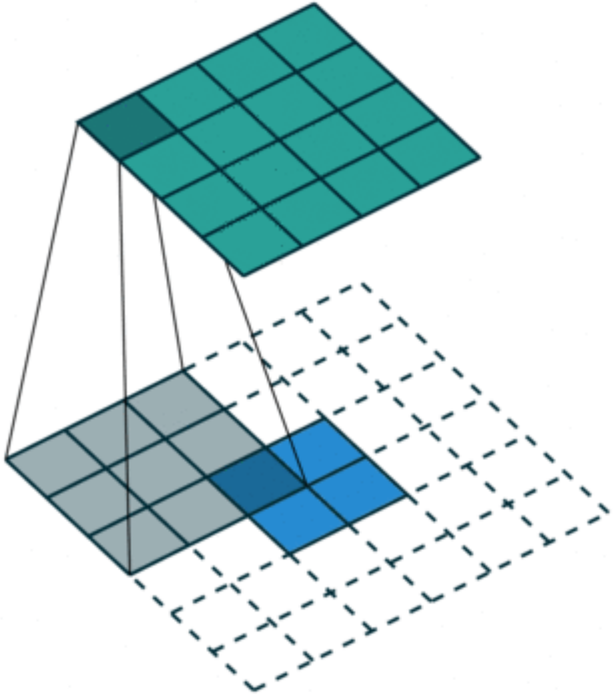
FCN /MSCOCO
Instance-ware Segm.

Credit: Li/Karpathy/Johnson

Learnable Upsampling: Deconvolution



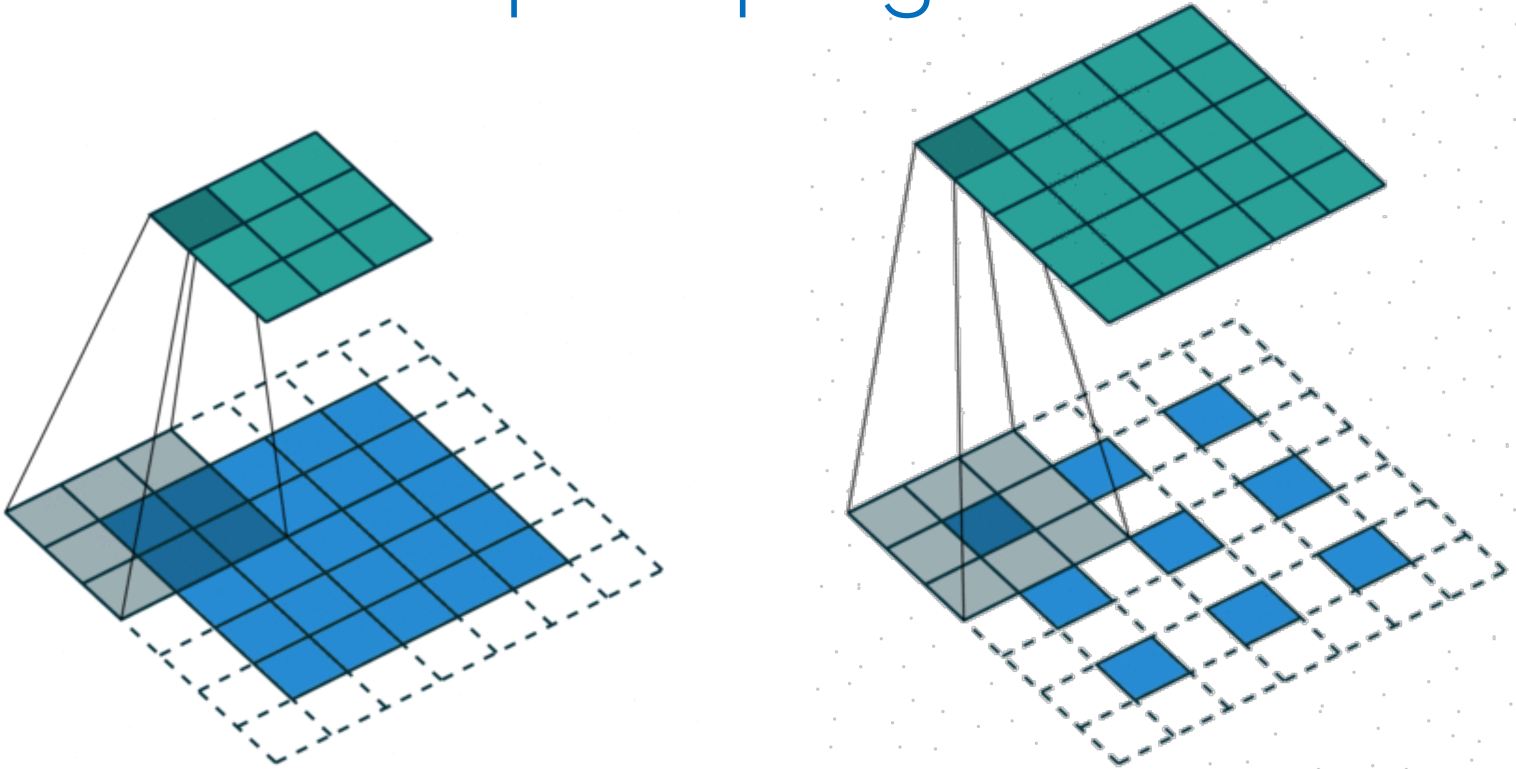
Convolution
no padding, no stride



Transposed convolution
no padding, no stride

https://github.com/vdumoulin/conv_arithmetic

Learnable Upsampling: Deconvolution

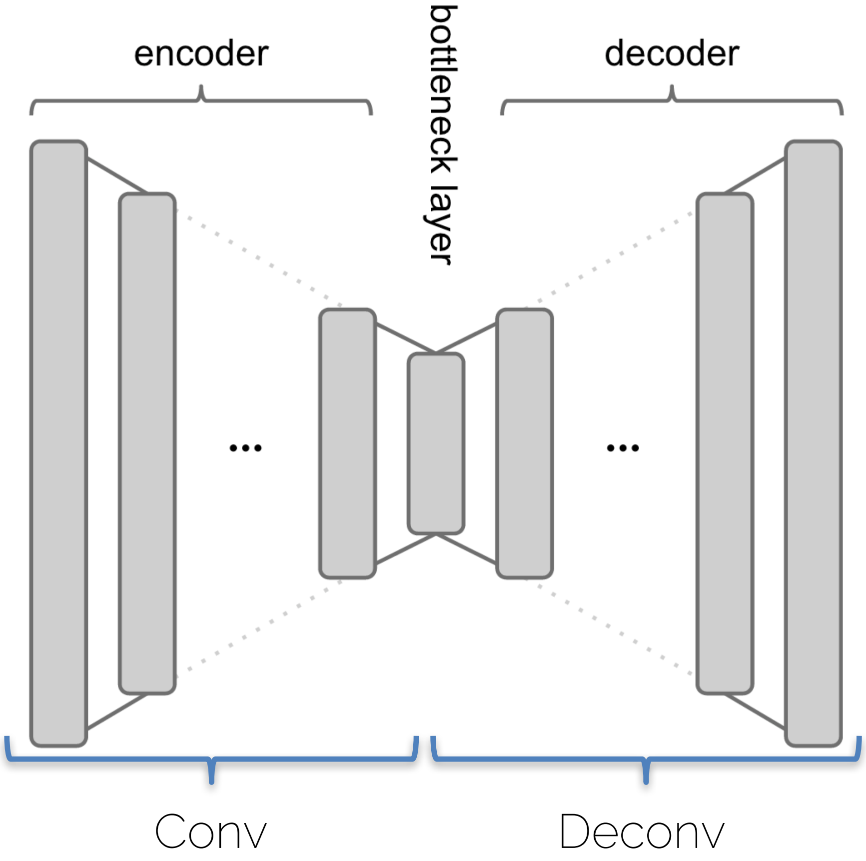


Convolution
padding, stride

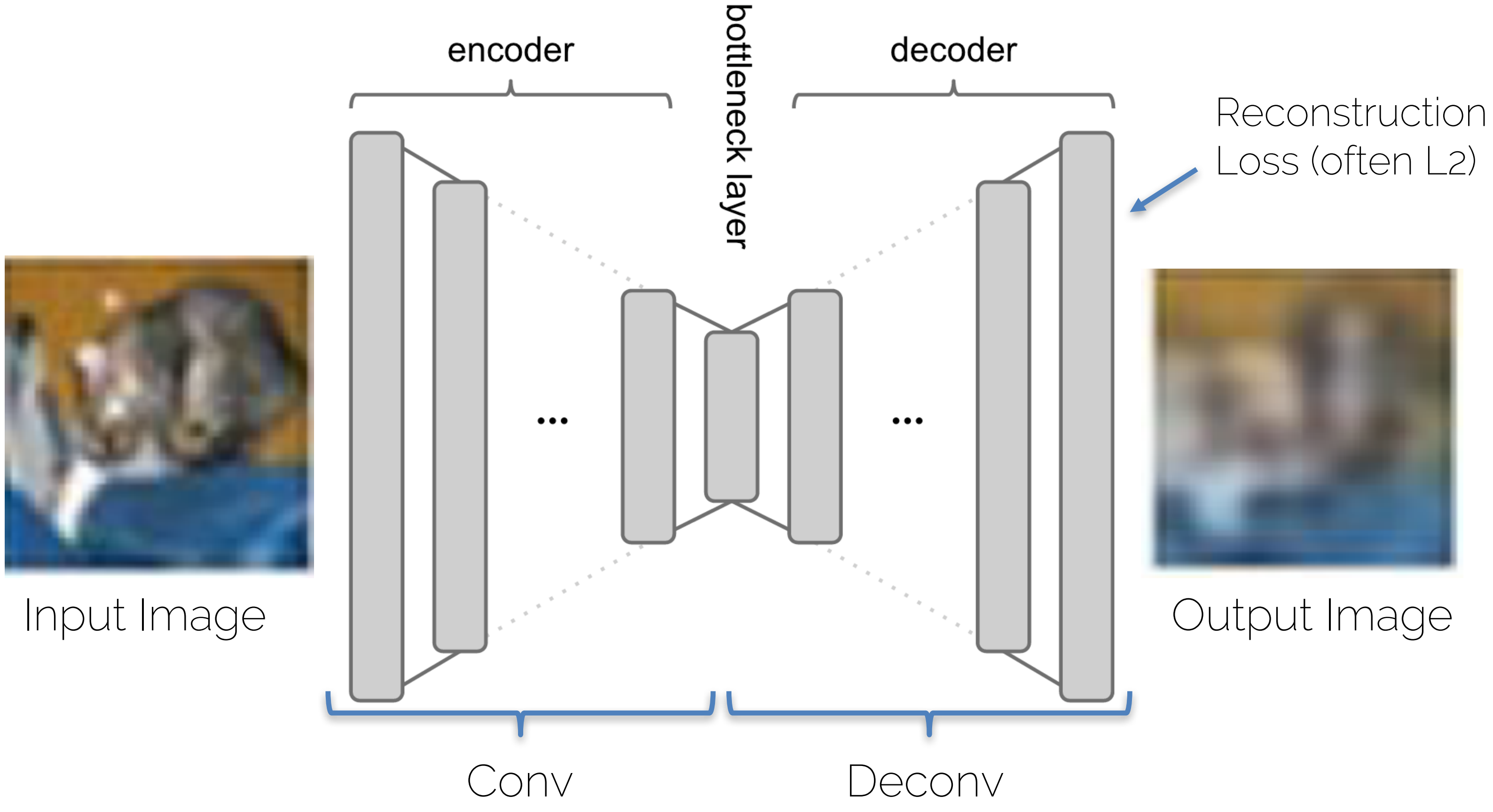
Transposed convolution
padding, stride

https://github.com/vdumoulin/conv_arithmetic

Autoencoder



Reconstruction: Autoencoder

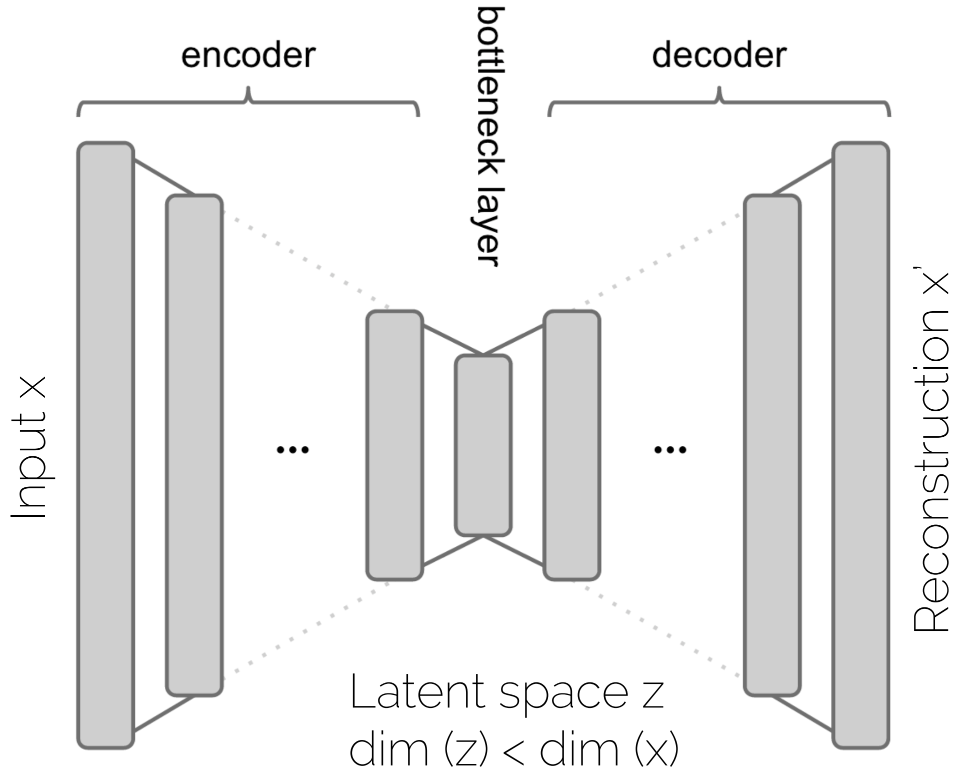


Training Classifiers vs Autoencoders

- Supervised Learning
  - Data (x, y)
    x is data, y is label
  - Goal: learn mapping x -> y
  - Example: classifier

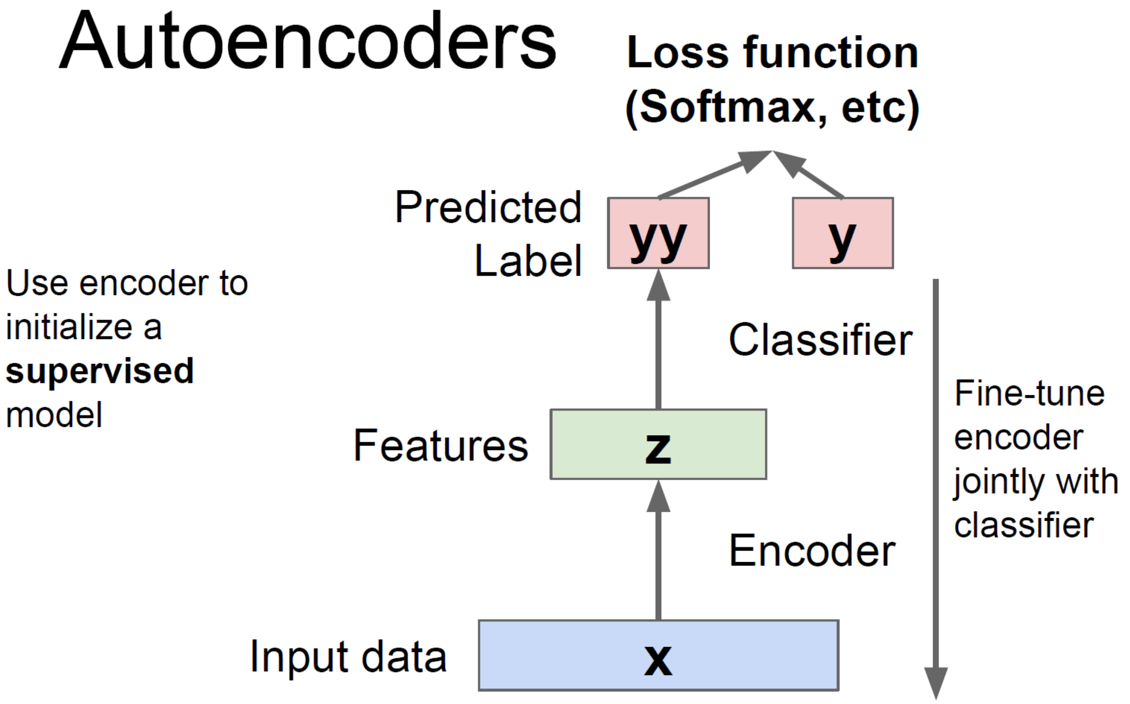
- Unsupervised Learning
  - Data (x)
    only data, no labels
  - Goal: learn structure (e.g., clustering)
  - Example: AE (autoencoder)

Training Autoencoders



Autoencoder: Use Cases

- Clustering
- Feature learning
- Embeddings

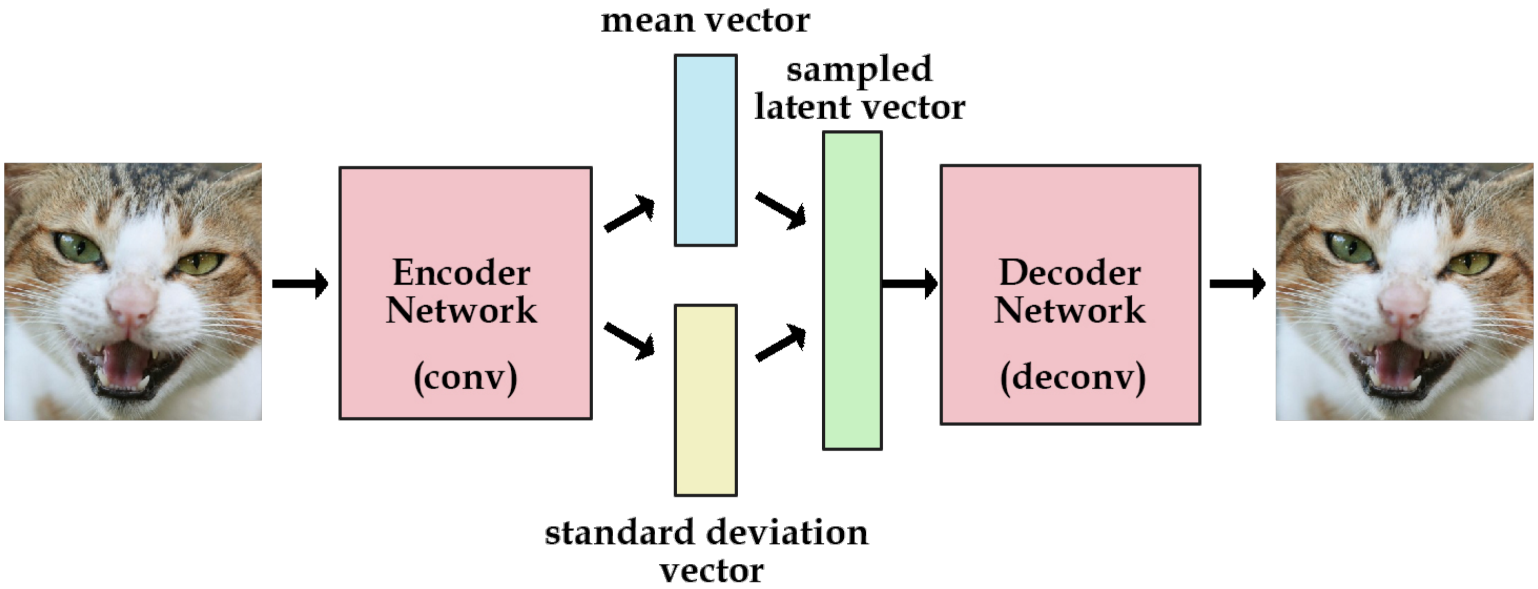


Pre-train AE -> fine-tune with small labeled data

figure by Li/Karpathy/Johnson

Generative Models

Variational Autoencoders (VAE)



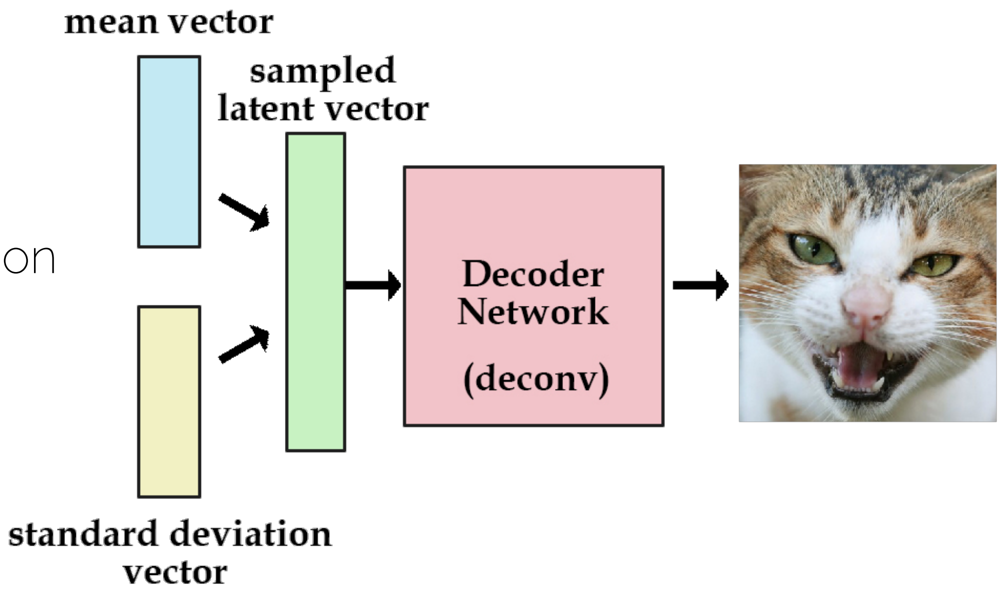
KL-Div Loss in latent space, forcing a unit Gaussian distribution
-> now the latent vector becomes a distribution

http://kvfrans.com/variational-autoencoders-explained/

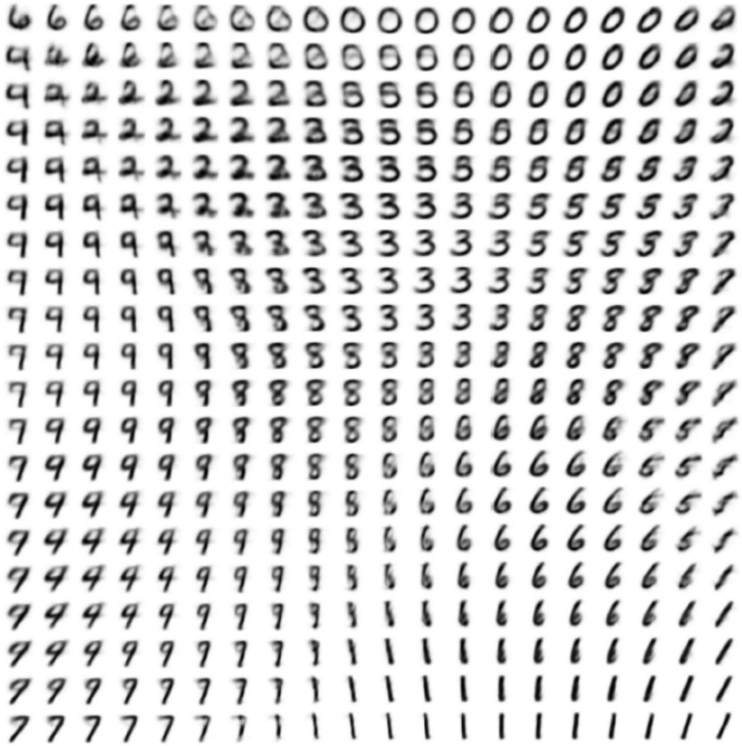
Variational Autoencoders (VAE)

- After training, generate random samples

Sample from the distribution
(e.g., unit Gaussian



Variational Autoencoders (VAE)



Autoencoder vs Variational Autoencoder

Autoencoder

Variational Autoencoder

Ground Truth

https://github.com/kvfrans/variational-autoencoder

Autoencoder Overview

- Autoencoders (AE)
  - Reconstruct input
  - Unsupervised learning
  - Latent space features are useful

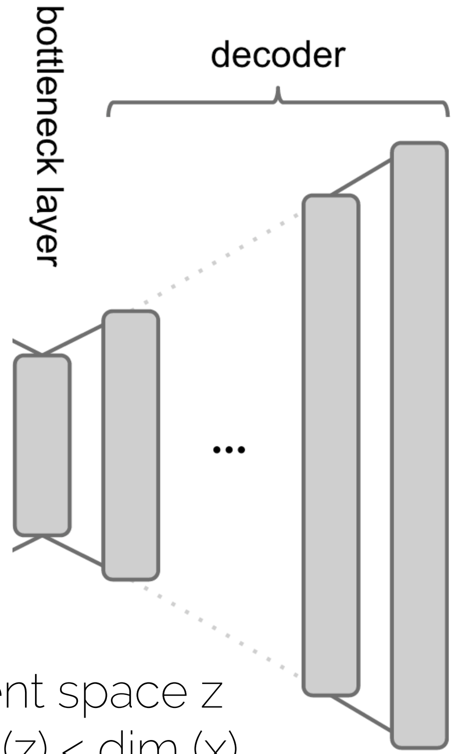
- Variational Autoencoders (VAE)
  - Probability distribution in latent space (e.g., Gaussian)
  - Sample from model to generate output

Discriminative vs Generative Tasks

- Discriminative Tasks:
  - Classification
  - Localization / Detection
  - Matching
  - Low-dimensional output
- Generative Tasks (more next lecture!)
  - Generate images / videos /shapes
  - High-dimensional output

Generative Models

"Test time":
-> reconstruction from
'random' vector



Latent space z
dim (z) < dim (x)

Reconstruction
Loss (often L2)

Output Image

Generative Models

- Pretty hard because of high-dim output
- Tends to “average” over training data
- Naïve variations don’t work in general settings

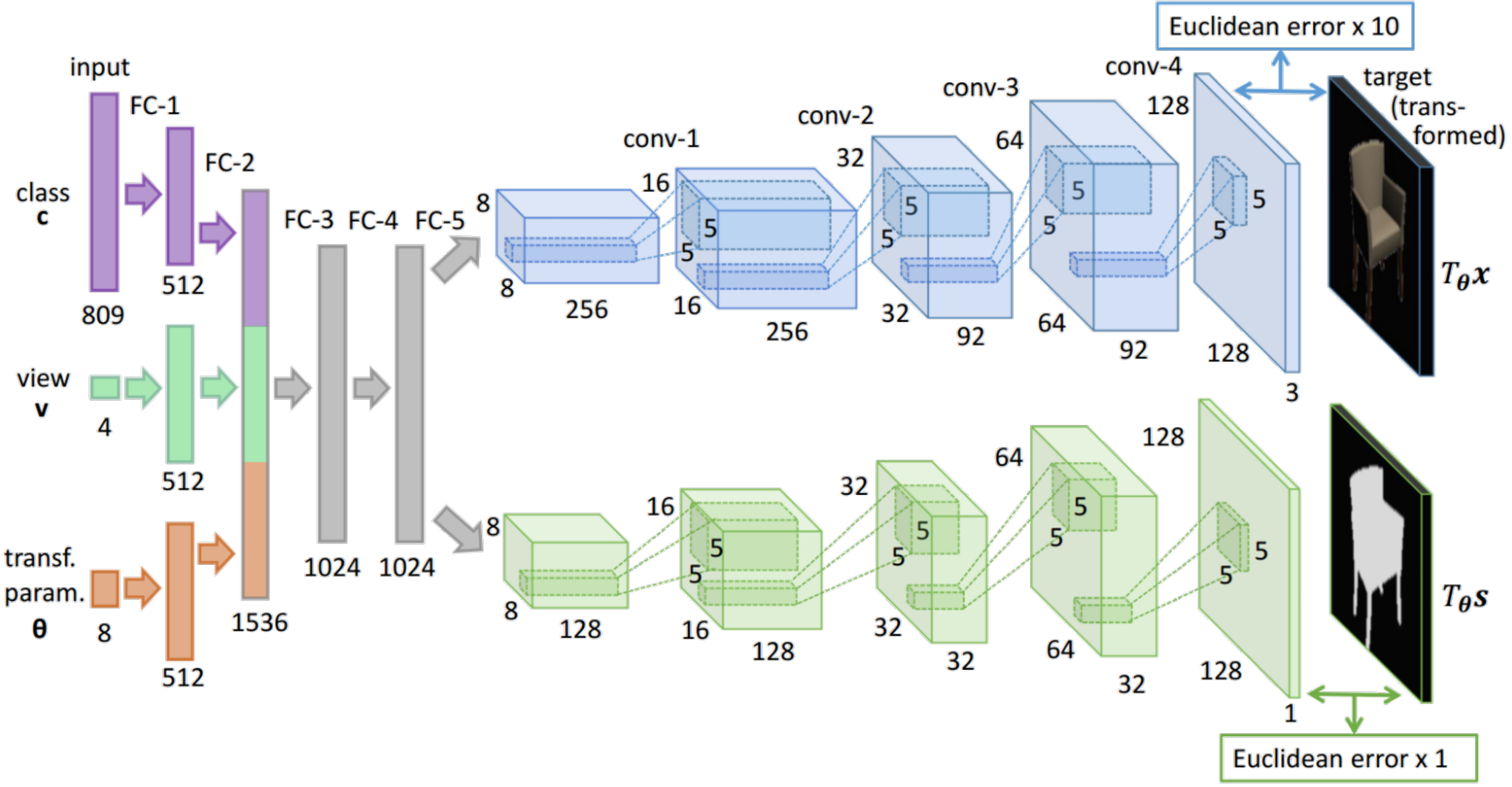
E.g., reconstructed images

Generative Models

- Probabilistic approaches help
  - Variational Autoencoders (VAE); e.g., [Kingma and Welling 13]
  - Deep belief networks; e.g., [Hinton et al. 06], [Lee et al. 09]

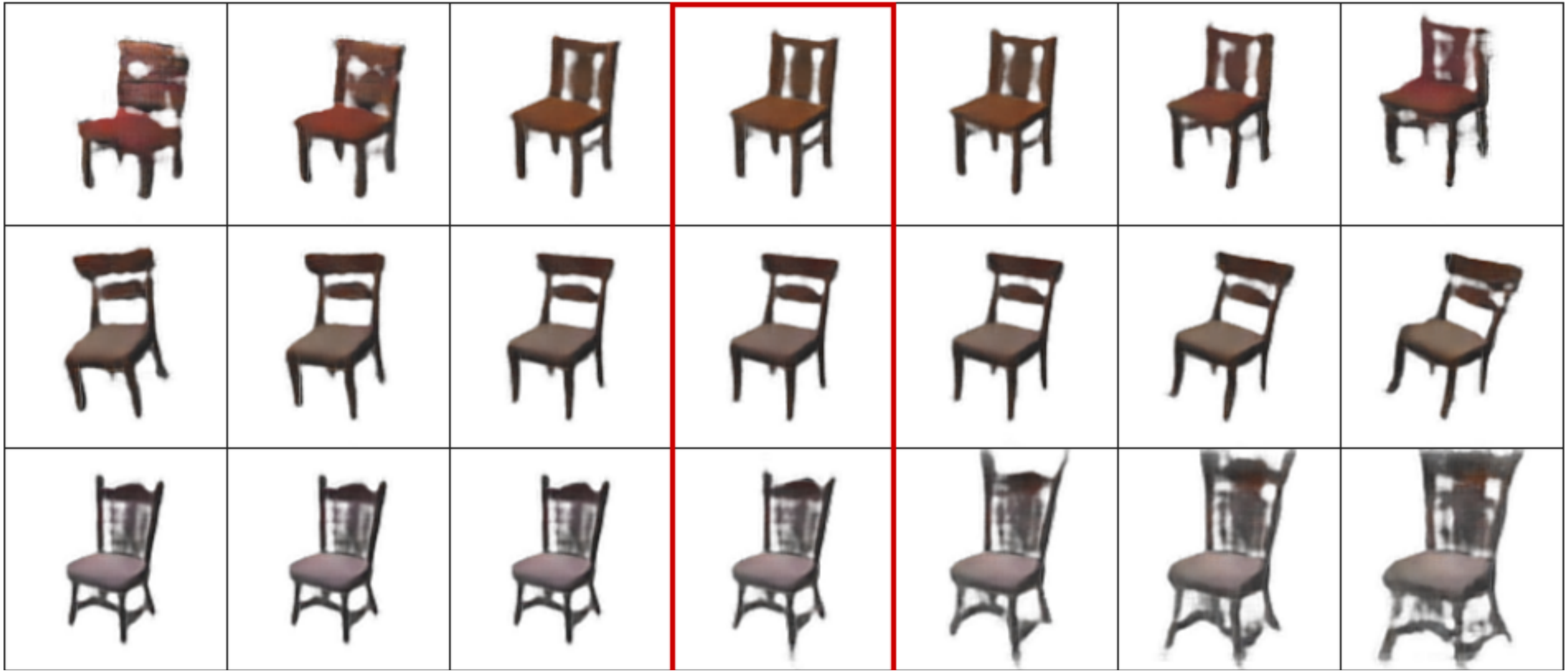
- Make problem easier
  - Domain-specific; e.g., chairs, faces, etc.

Generative Models



[Dosovitsky et al. 14] Learning to Generate Chairs

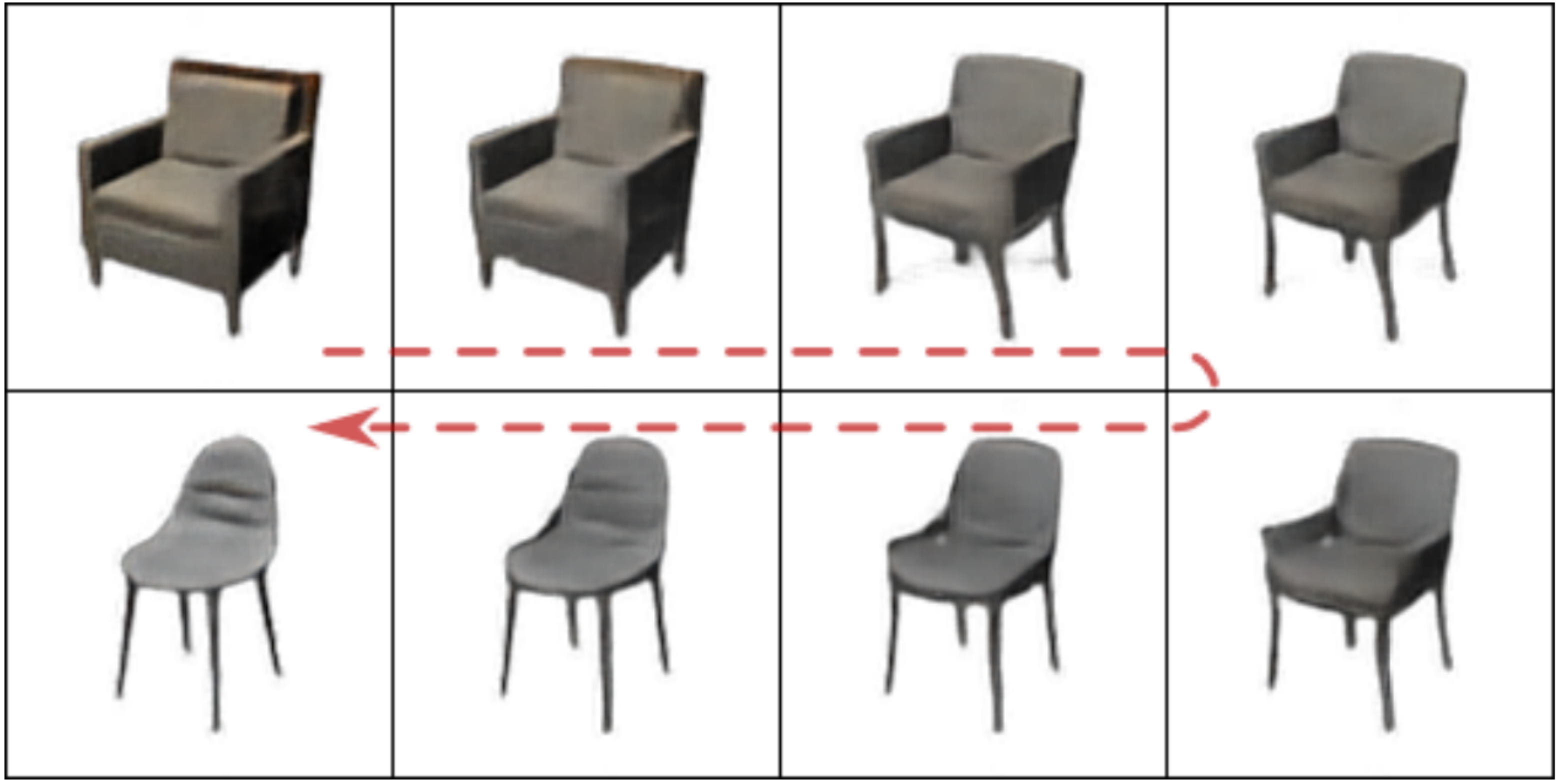
Generative Models



Generation of chair images
while activating various transforms

[Dosovitsky et al. 14] Learning to Generate Chairs

Generative Models



Interpolation between two chair models

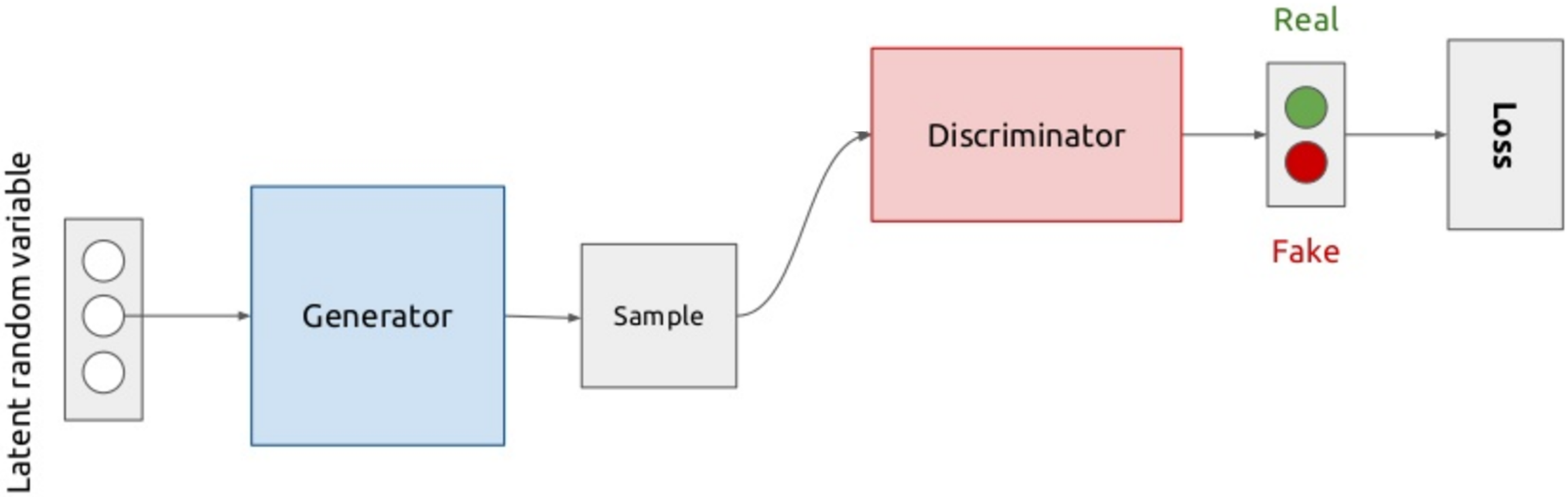
[Dosovitsky et al. 14] Learning to Generate Chairs

Generative Models

Morphing between
chair models

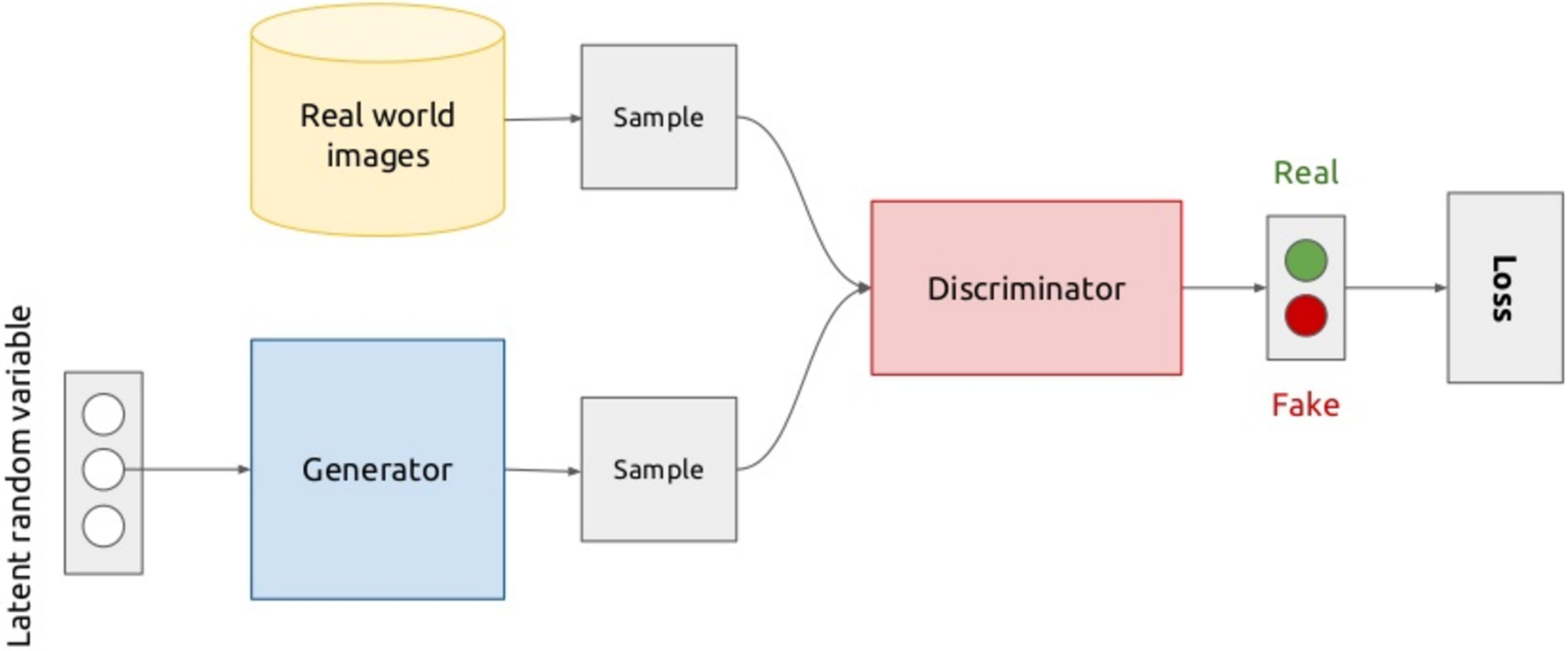
[Dosovitsky et al. 14] Learning to Generate Chairs

Generative Adversarial Networks (GANs)



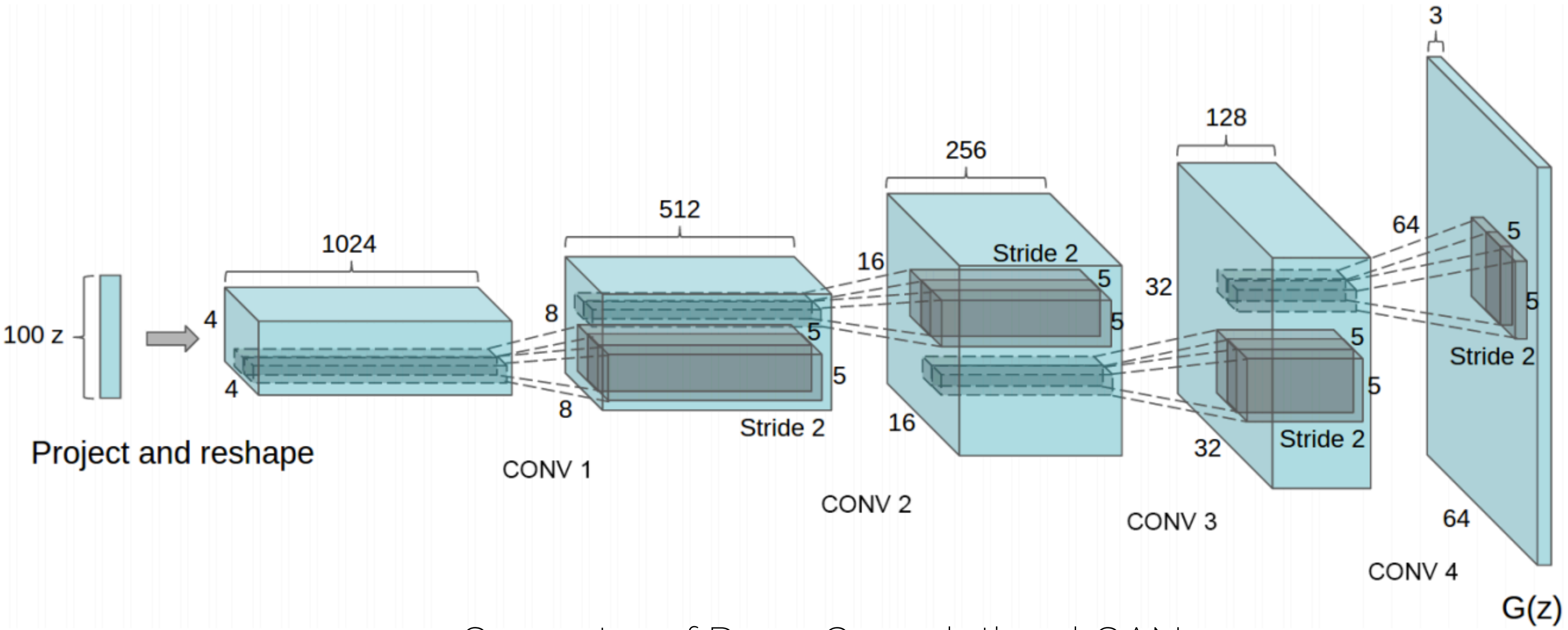
[Goodfellow et al. 14] GANs (slide McGuinness)

Generative Adversarial Networks (GANs)



[Goodfellow et al. 14] GANs (slide McGuinness)

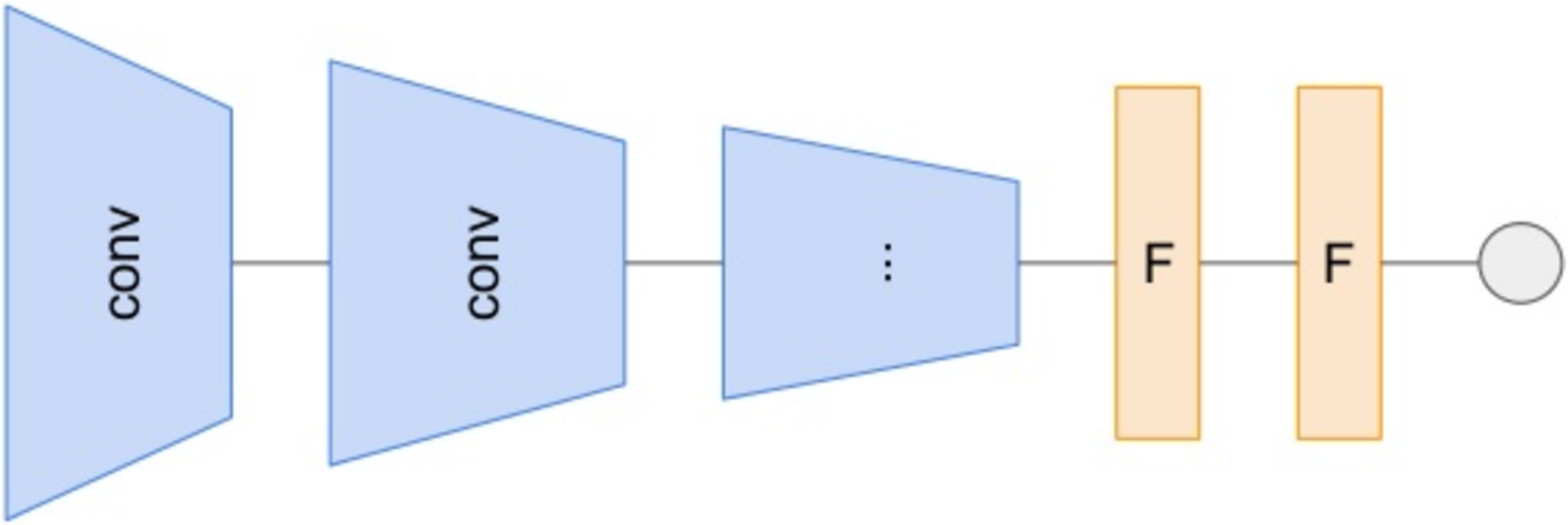
GANs: Generator



Generator of Deep Convolutional GANs

DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

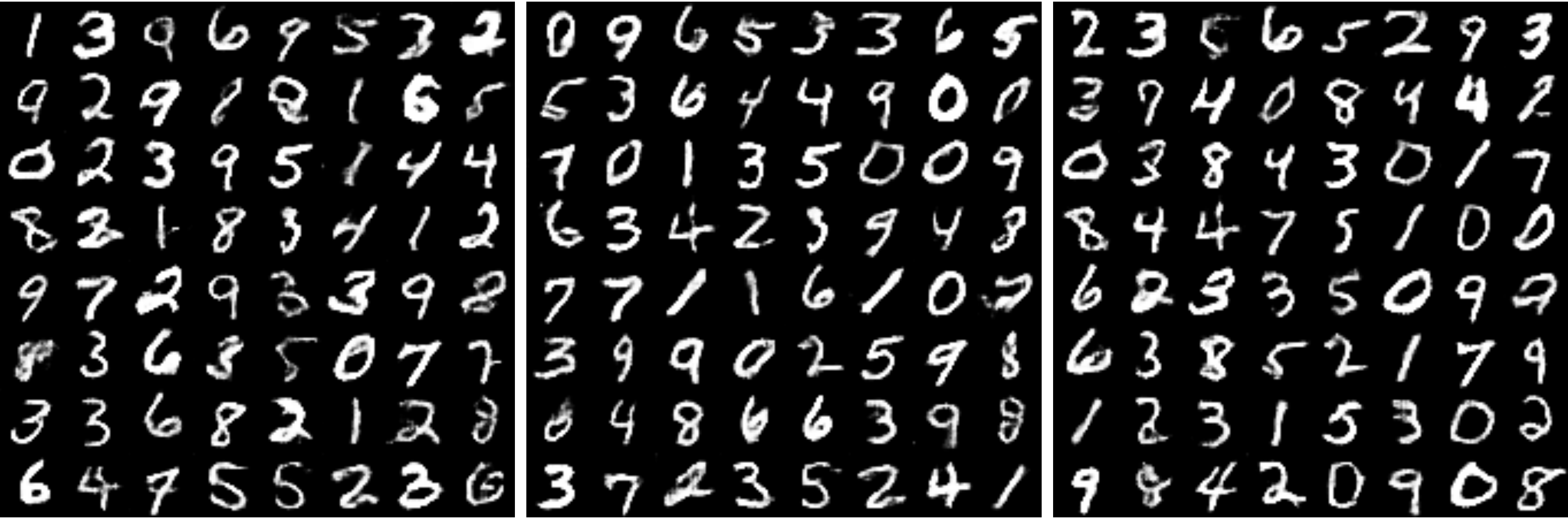
GANs: Discriminator



Tries to distinguish between real and fake input

[Goodfellow et al. 14] GANs (slide McGuinness)

DCGAN: Results



Results on MNIST

DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

DCGAN: Results

Results on CelebA (200k relatively well aligned portrait photos)

DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

DCGAN: Results

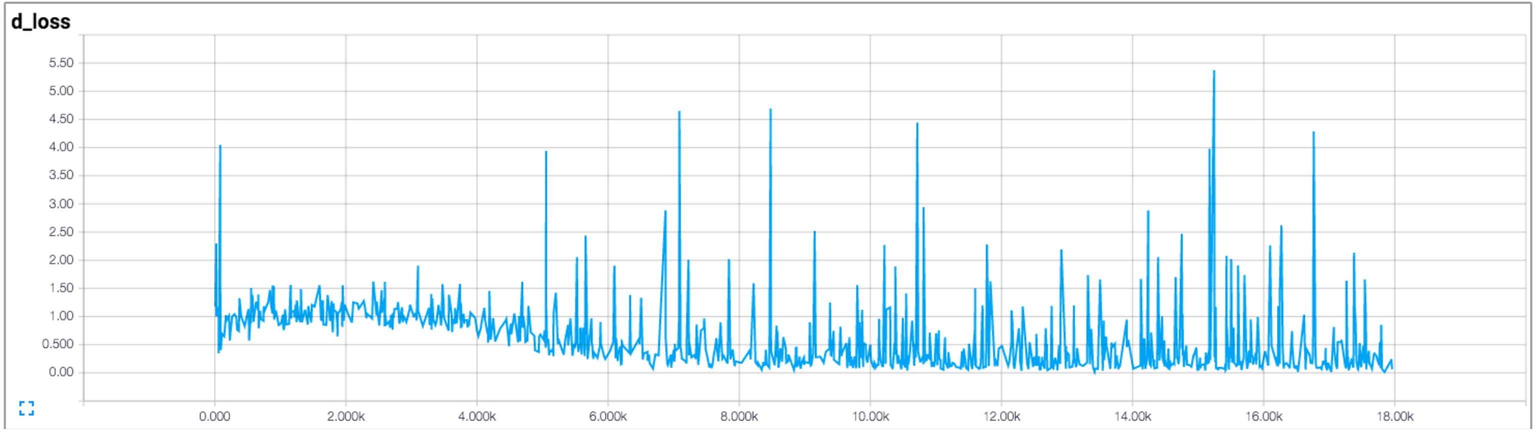
Asian face dataset

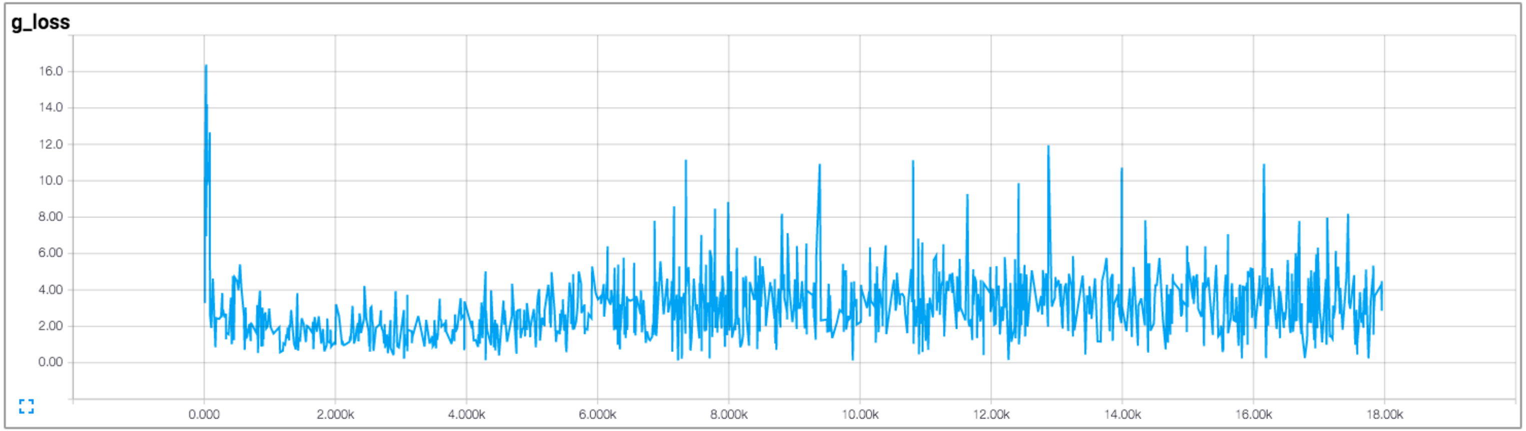
DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

DCGAN: Results

DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

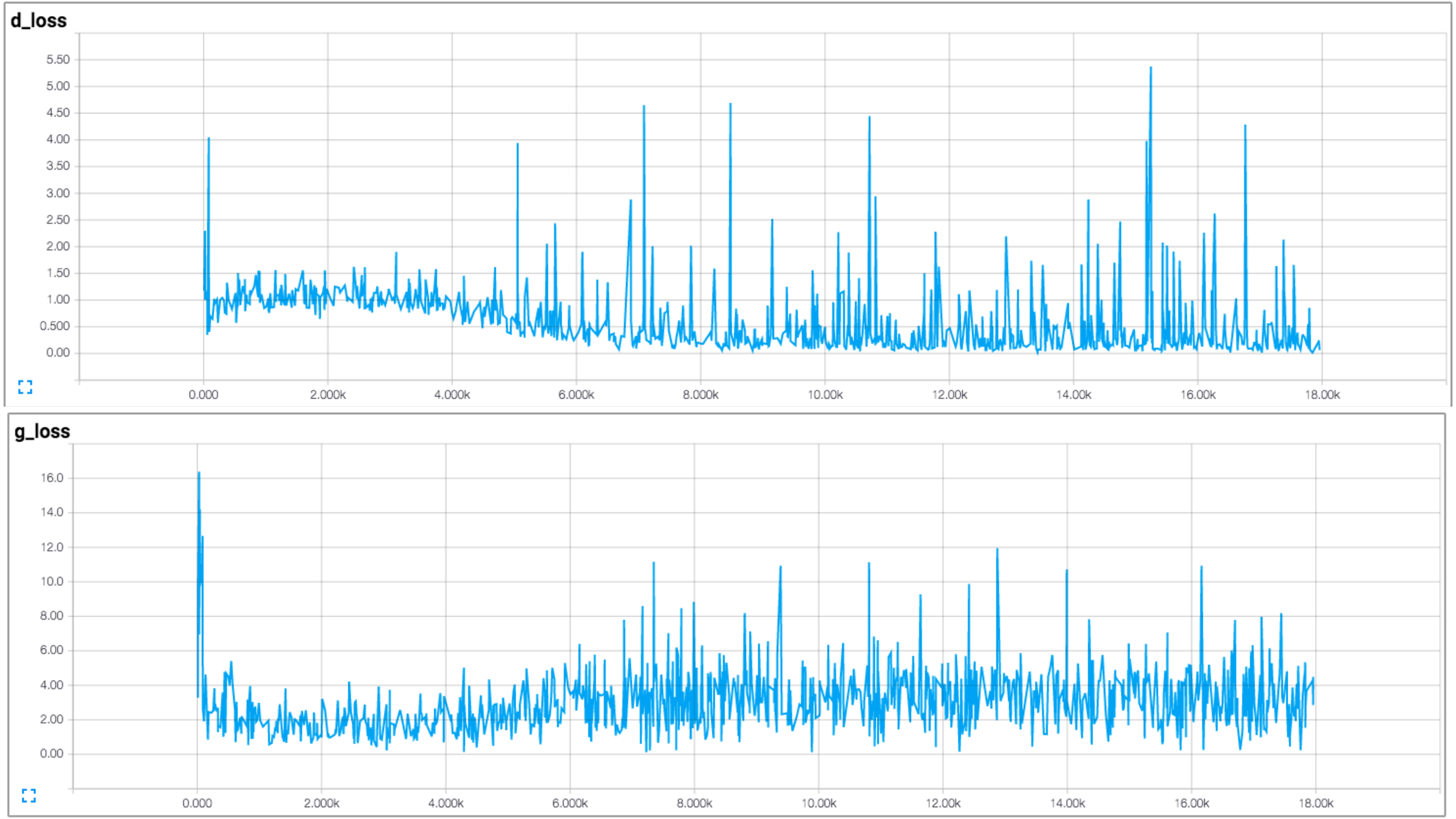
DCGAN: Results





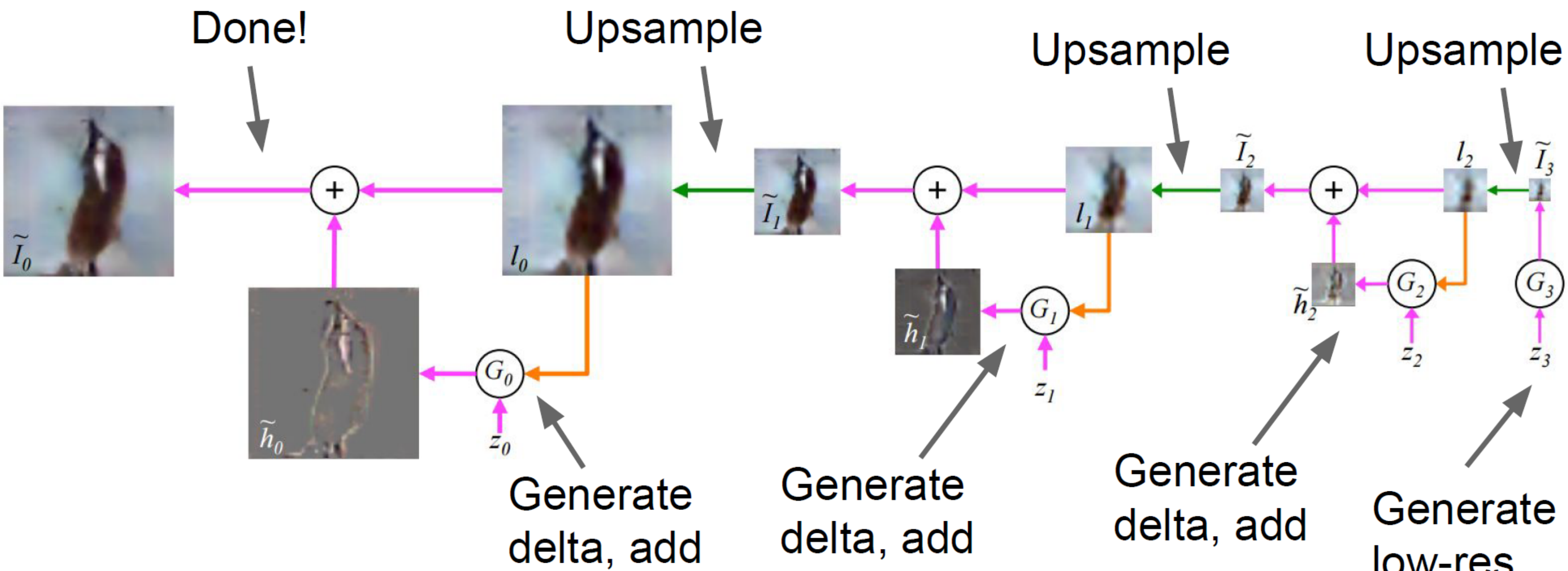
DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

DCGAN: Results



DCGAN: https://github.com/carpedm20/DCGAN-tensorflow

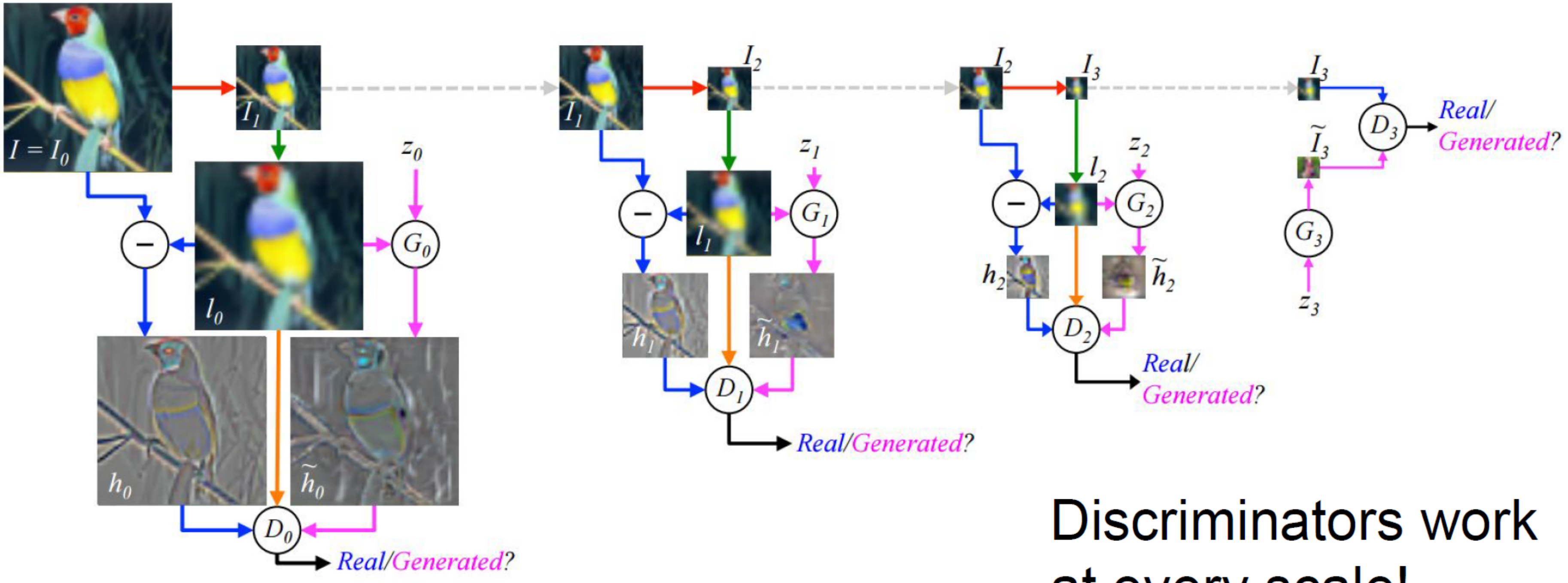
Lots of GAN Variations: E.g., Multiscale



Denton et al, "Deep generative image models using a Laplacian pyramid of adversarial networks", NIPS 2015

Credit: Li/Karpathy/Johnson

Lots of GAN Variations: E.g., Multiscale

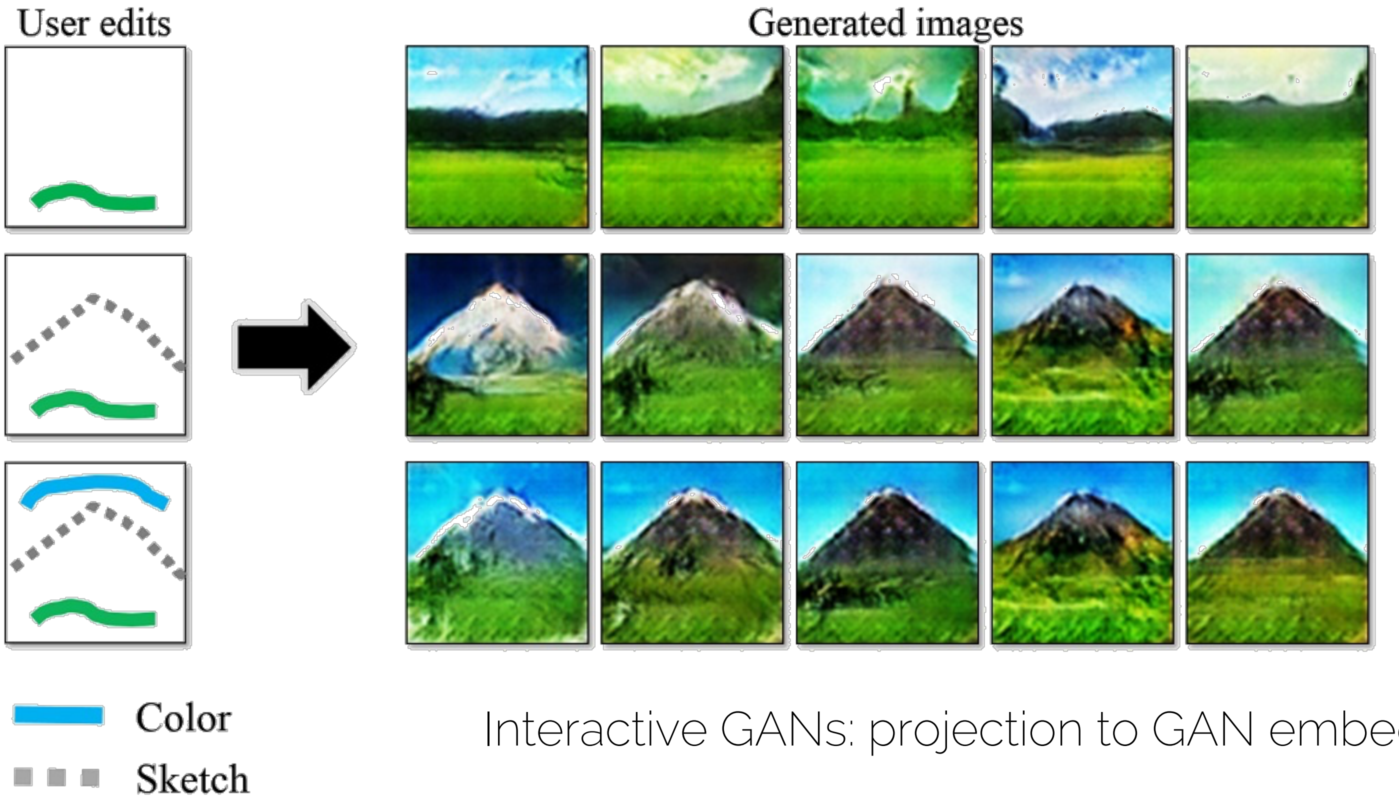


Discriminators work at every scale!

Denton et al, NIPS 2015

Credit: Li/Karpathy/Johnson

Lots of GAN Variations: E.g., iGAN



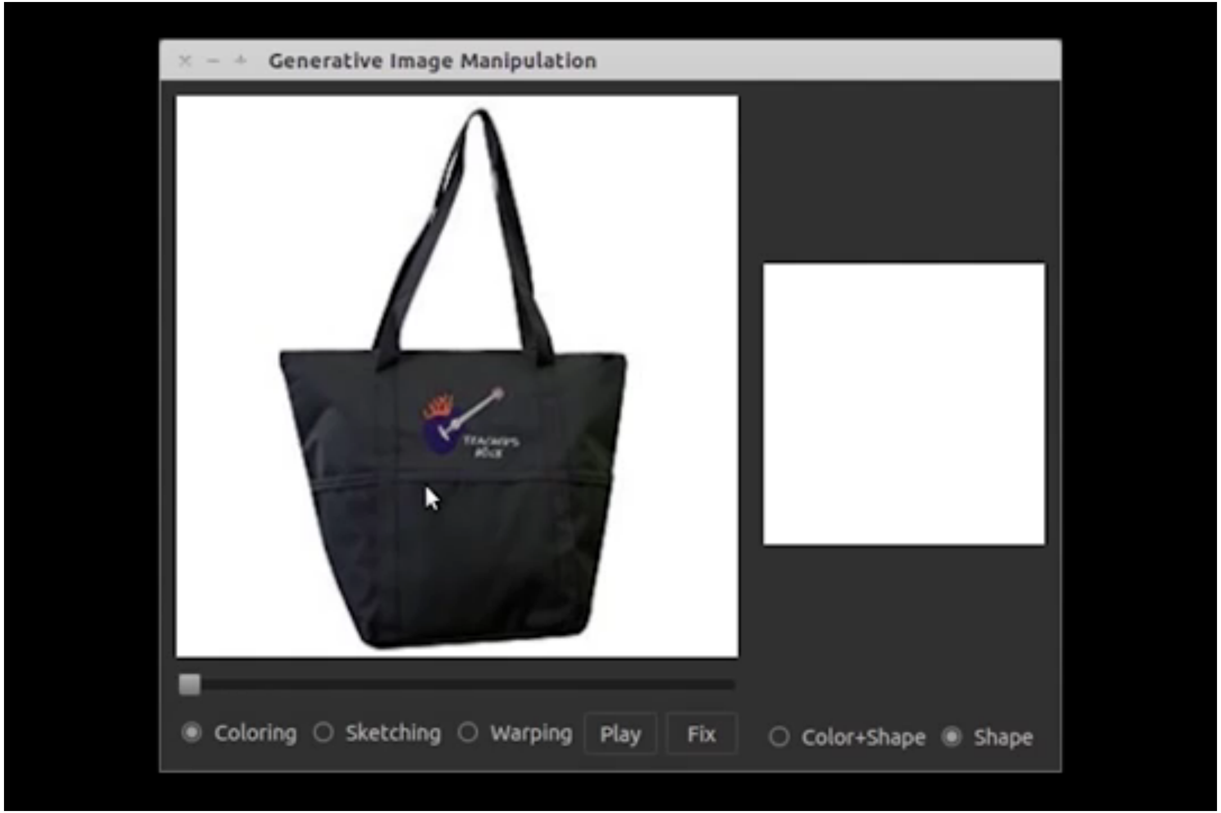
Interactive GANs: projection to GAN embedding

https://github.com/junyanz/iGAN [Zhu et al. 16.]

Lots of GAN Variations: E.g., iGAN

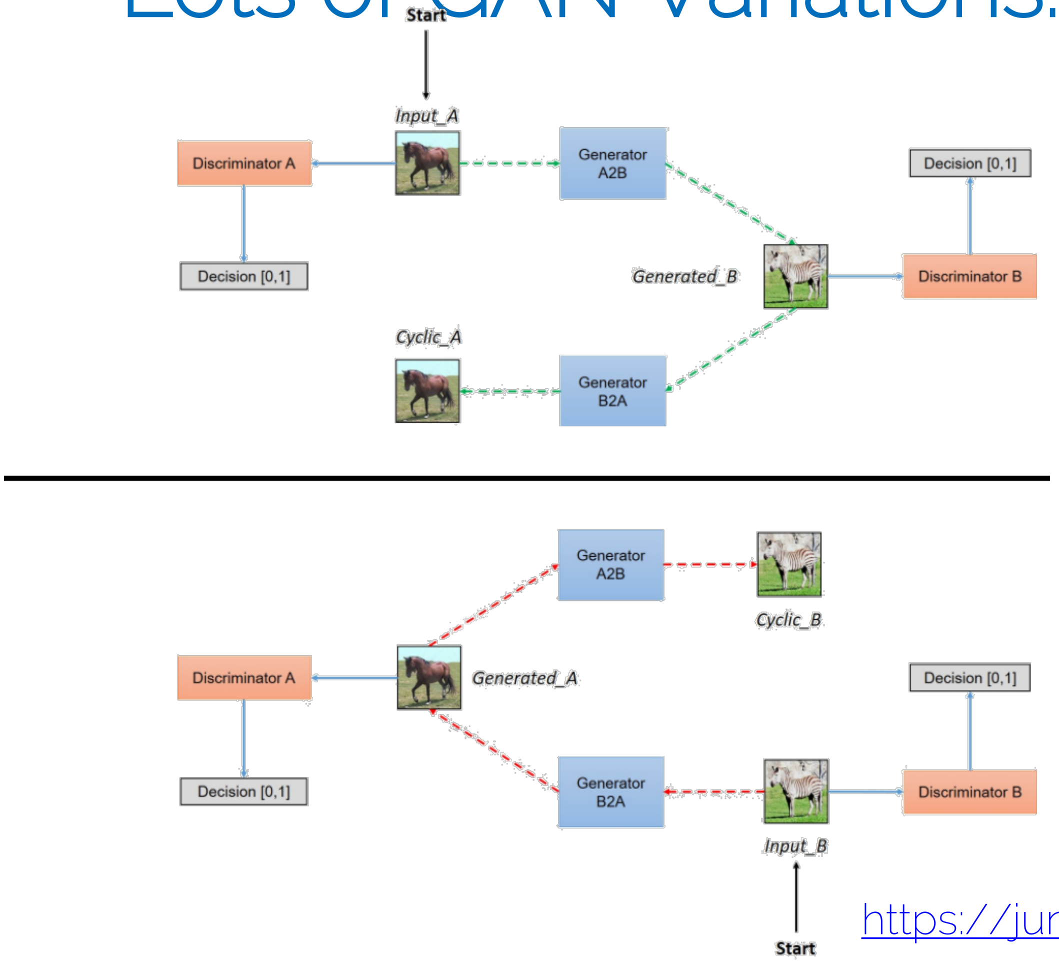
https://github.com/junyanz/iGAN [Zhu et al. 16.]

Lots of GAN Variations: E.g., iGAN



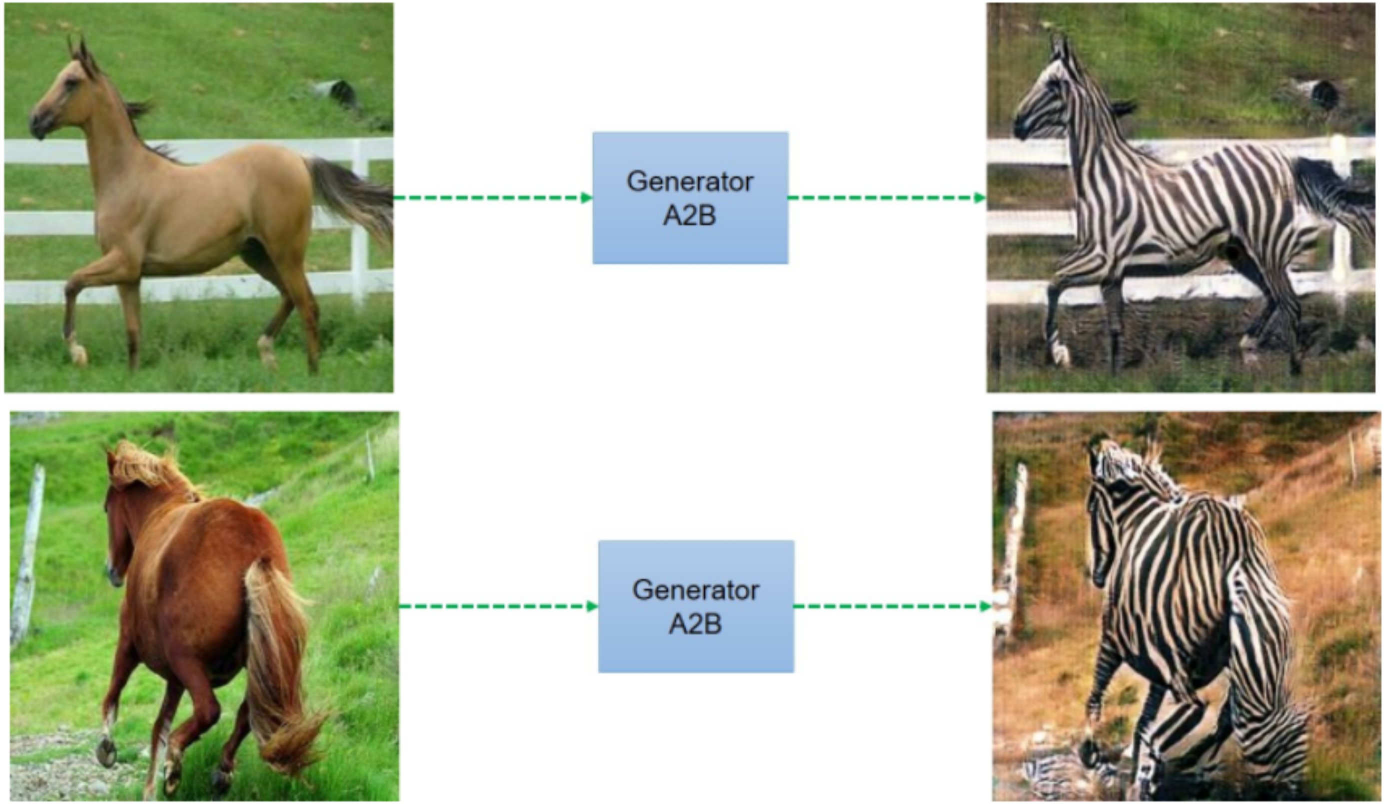
https://github.com/junyanz/iGAN [Zhu et al. 16.]

Lots of GAN Variations: E.g., Cycle GAN



https://junyanz.github.io/CycleGAN/ [Zhu et al. 17.]

Lots of GAN Variations: E.g., Cycle GAN



https://junyanz.github.io/CycleGAN/ [Zhu et al. 17.]

Lots of GAN Variations: E.g., Cycle GAN

Does not always work ☺

https://junyanz.github.io/CycleGAN/ [Zhu et al. 17.]

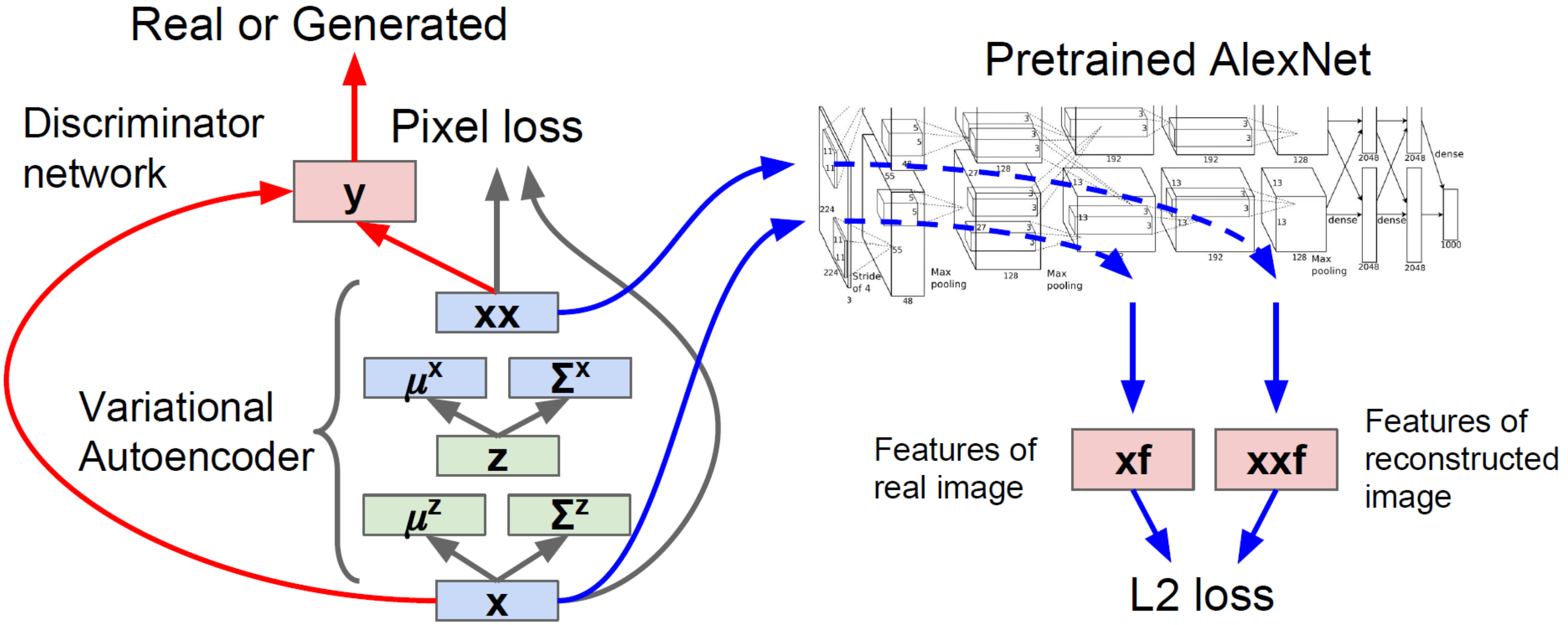
GANs: Still Open Problem!

- Pretty hard in the general case: e.g., CIFAR-10

Nearest neighbor form training set

Credit: Li/Karpathy/Johnson

GAN + VAE

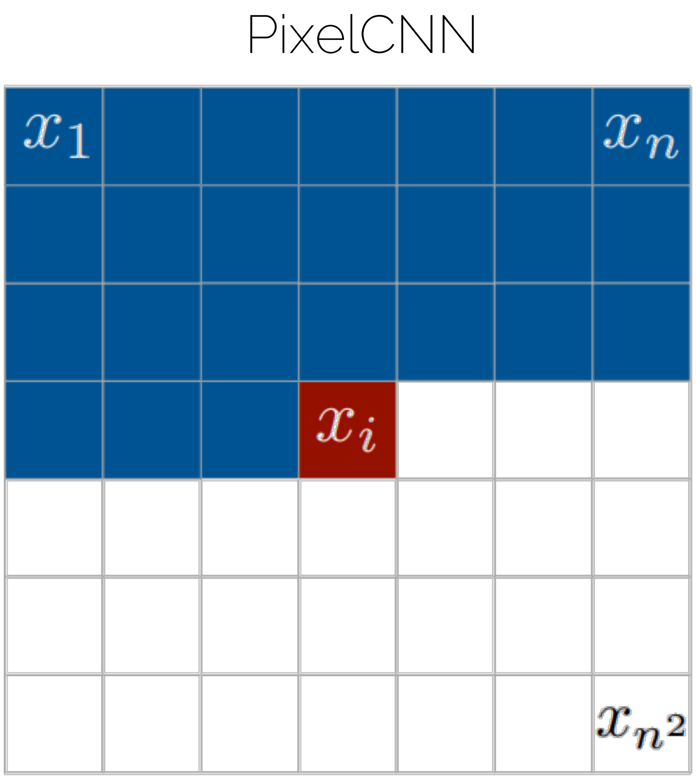


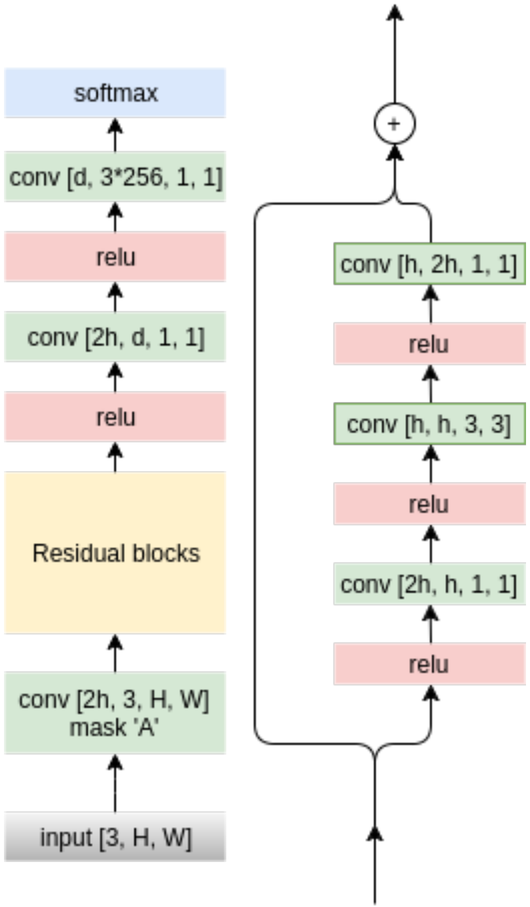
[Dosovitskiy and Brox 16] (slide Li/Karpathy/Johnson)

GAN + VAE

[Dosovitskiy and Brox 16] (slide Li/Karpathy/Johnson)

Other Generative Models: PixelCNN





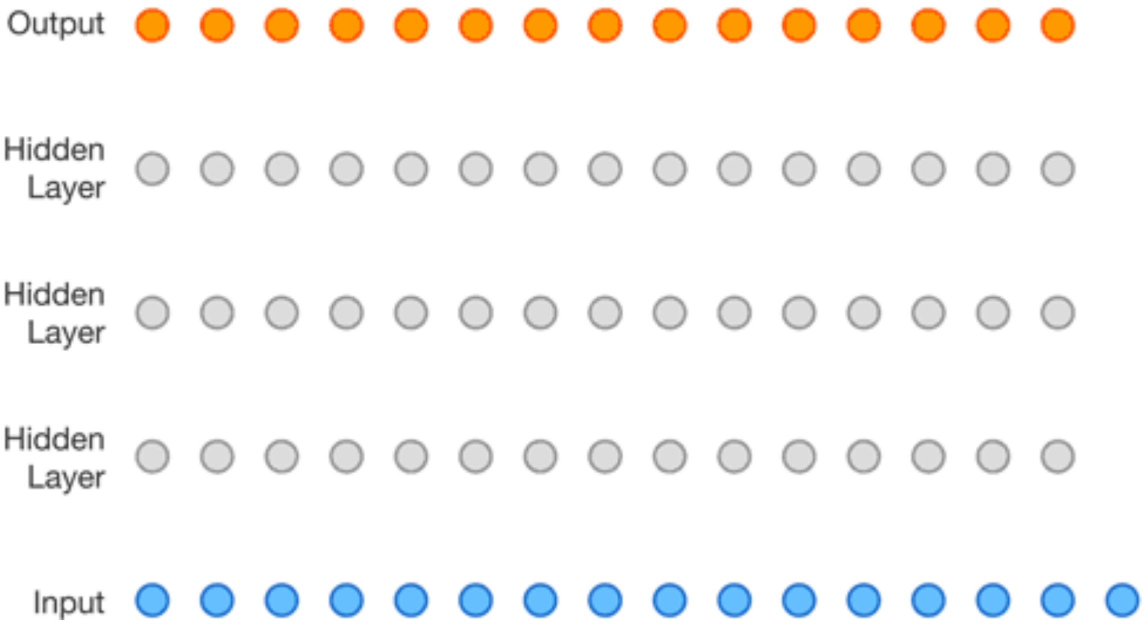
[van den Oord 16] PixelCNN (slide: http://sergeiturukin.com/2017/02/22/pixelcnn.html

Other Generative Models: PixelCNN

Left: source images; right: new portraits generated from high-level latent representation

[van den Oord 16] PixelCNN

Other Generative Models: WaveNet



[van der Ooord 16] https://deepmind.com/blog/wavenet-generative-model-raw-audio/

Challenges with Conditional Chaining

- How to train?
  - We have ground images
  - But no ground truth for intermediate predictions
  - Typically very challenging and requires several passes
  - Really costly at test time!
    - E.g., Audio is 16k samples / second -> 16k forward passes…

Multi-Dimensional ConvNets

Multi-Dimensional ConvNets

- 1D ConvNets
  - Audio / Speech
  - Also Point Clouds
- 2D ConvNets
  - Images (AlexNet, VGG, ResNet -> Classification, Localization, etc..)
- 3D ConvNets
  - For videos
  - For 3D data
- 4D ConvNets
  - E.g., dynamic 3D data (Haven't seen much work there)
  - Simulations

What are Convolutions?

Discrete case: box filter

f	**4**	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**

g	**1/3**	**1/3**	**1/3**

'Slide' filter kernel from left to right; at each position, compute a single value in the output data

Remember: 1D Convolutions

Discrete case: box filter

f	**4**	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**
g	**1/3**	**1/3**	**1/3**							
$f * g$		**3**								

$$4 \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} = 3$$

Remember: 1D Convolutions

Discrete case: box filter

f	**4**	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**
g		**1/3**	**1/3**	**1/3**						
$f * g$		**3**	**0**							

$$3 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} + (-5) \cdot \frac{1}{3} = 0$$

Remember: 1D Convolutions

Discrete case: box filter

f	4	3	2	-5	3	5	2	5	5	6
g			1/3	1/3	1/3					
$f * g$		3	0	0						

$$2 \cdot \frac{1}{3} + (-5) \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} = 0$$

Remember: 1D Convolutions

Discrete case: box filter

f	**4**	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**
g				**1/3**	**1/3**	**1/3**				
$f * g$		**3**	**0**	**0**	**1**					

$$(-5) \cdot \frac{1}{3} + 3 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} = 1$$

Remember: 1D Convolutions

Discrete case: box filter

f	4	3	2	-5	3	5	2	5	5	6
g					1/3	1/3	1/3			
$f * g$		3	0	0	1	10/3				

$$3 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} = \frac{10}{3}$$

Remember: 1D Convolutions

Discrete case: box filter

f	**4**	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**
g						**1/3**	**1/3**	**1/3**		
$f * g$		**3**	**0**	**0**	**1**	**10/3**	**4**			

$$5 \cdot \frac{1}{3} + 2 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} = 4$$

Remember: 1D Convolutions

Discrete case: box filter

f	**4**	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**
g							**1/3**	**1/3**	**1/3**	
$f * g$		**3**	**0**	**0**	**1**	**10/3**	**4**	**4**		

$$2 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} = 4$$

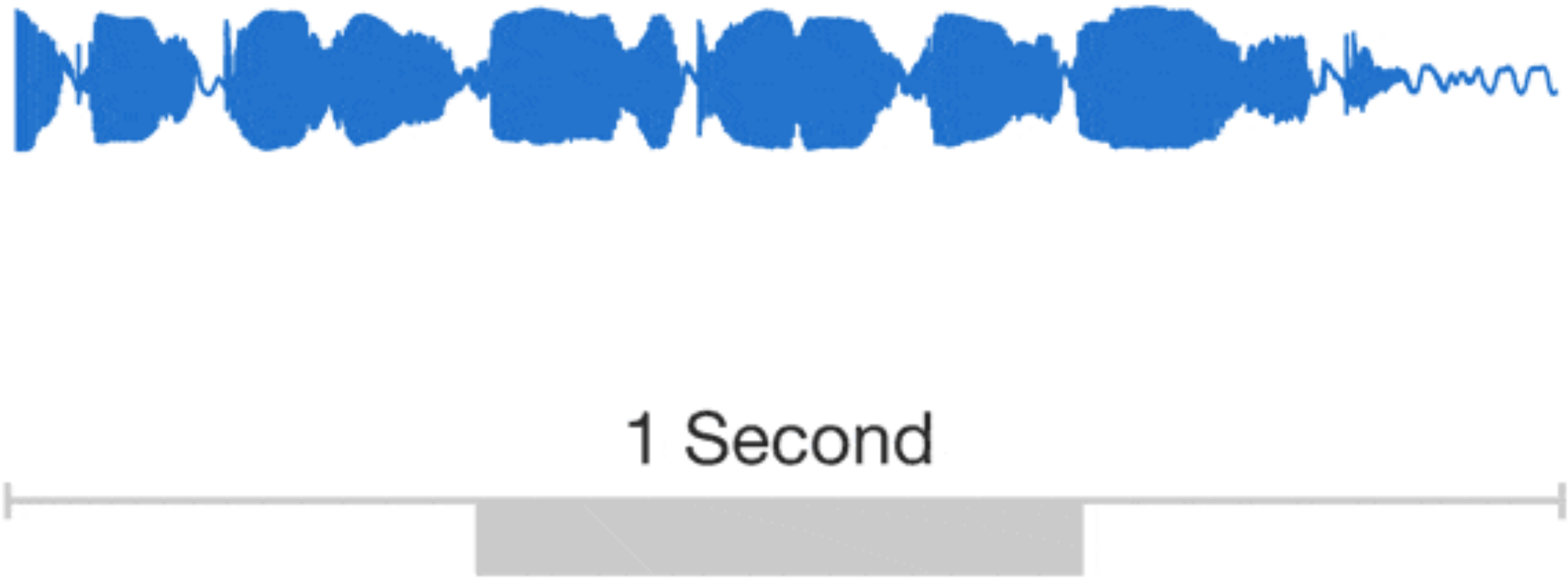
Remember: 1D Convolutions

Discrete case: box filter

f	**4**	**3**	**2**	**-5**	**3**	**5**	**2**	**5**	**5**	**6**
g								**1/3**	**1/3**	**1/3**
$f * g$		**3**	**0**	**0**	**1**	**10/3**	**4**	**4**	**16/3**	

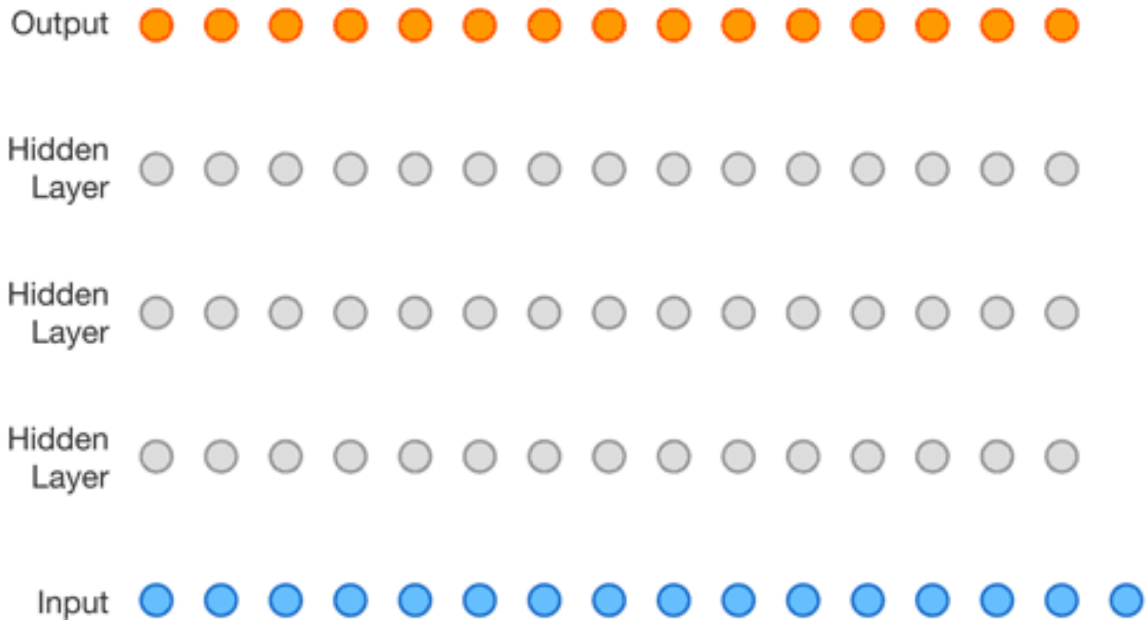
$$5 \cdot \frac{1}{3} + 5 \cdot \frac{1}{3} + 6 \cdot \frac{1}{3} = \frac{16}{3}$$

1D ConvNets: WaveNet



[van der Ooord 16] https://deepmind.com/blog/wavenet-generative-model-raw-audio/

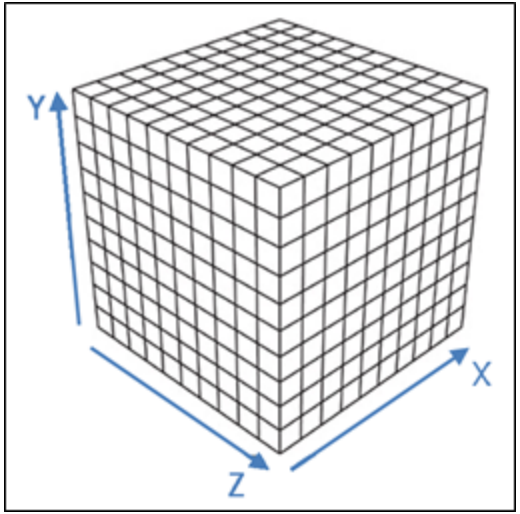
1D ConvNets: WaveNet



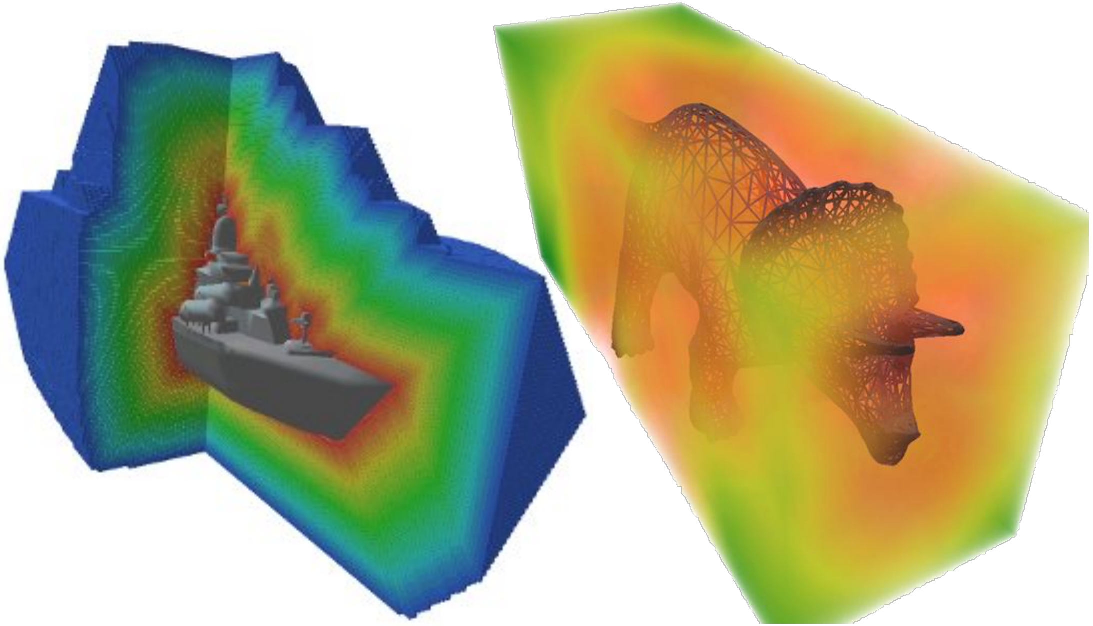
[van der Ooord 16] https://deepmind.com/blog/wavenet-generative-model-raw-audio/

3D Convolutions

On volumetric data structures

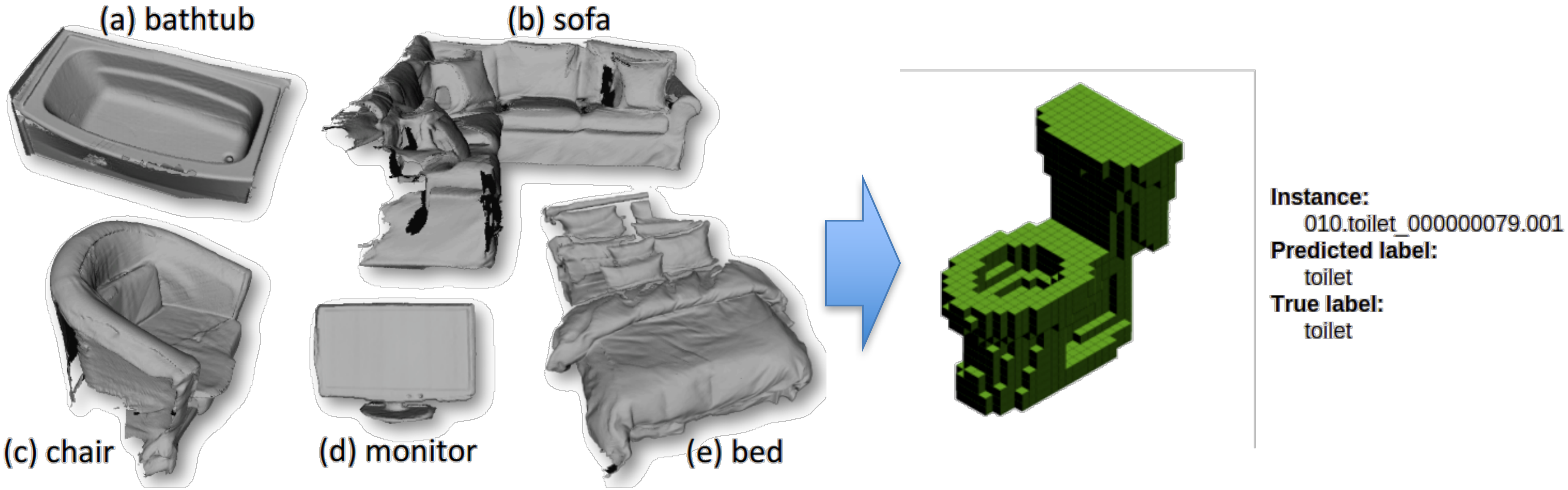


(binary) Voxel Grid



Implicit functions: e.g., signed distance field

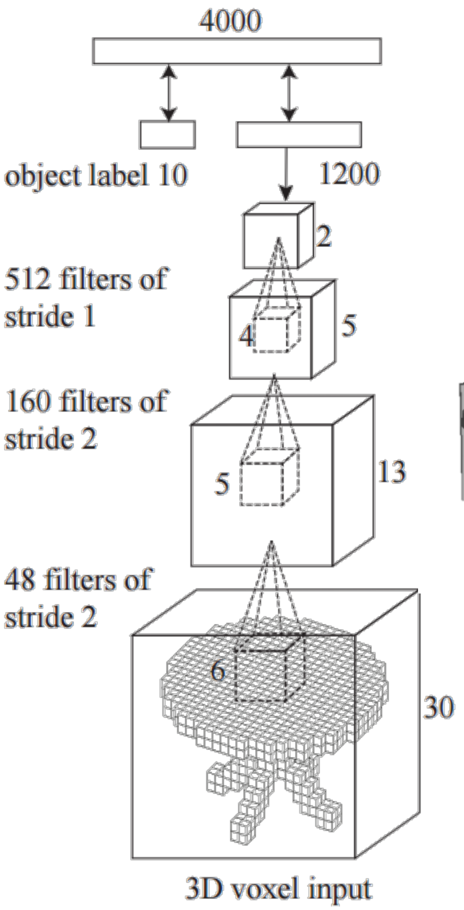
3D Classification

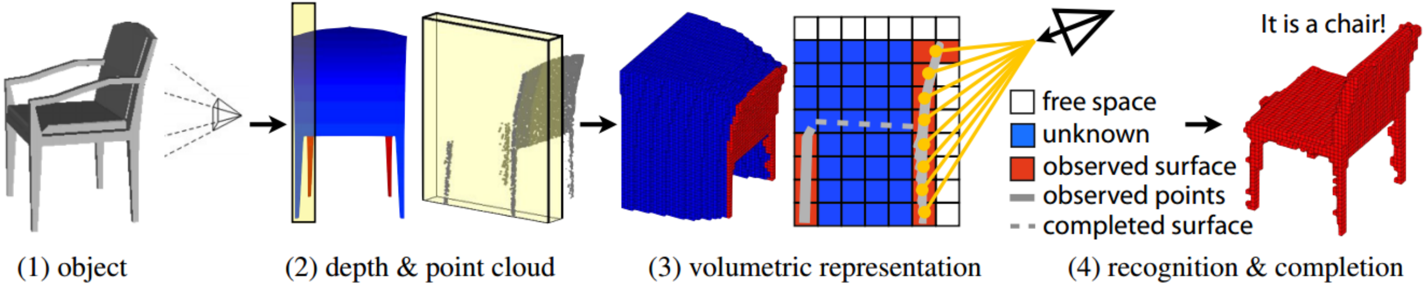


Class from 3D model (e.g., obtained with Kinect Scan)

[Maturana et al. 15] & [Qi et al. 16] 3D vs Multi-view

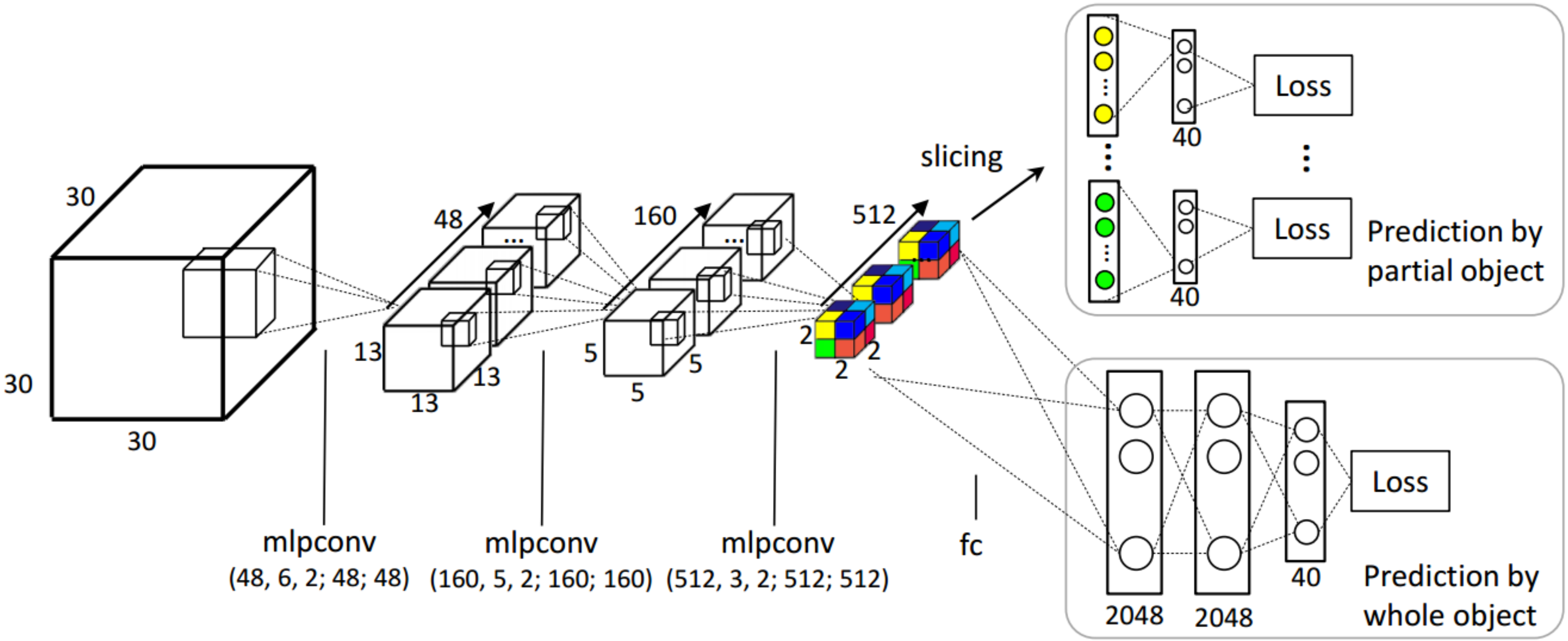
3D Classification





[Wu et al. 14] 3D ShapeNets

3D Classification



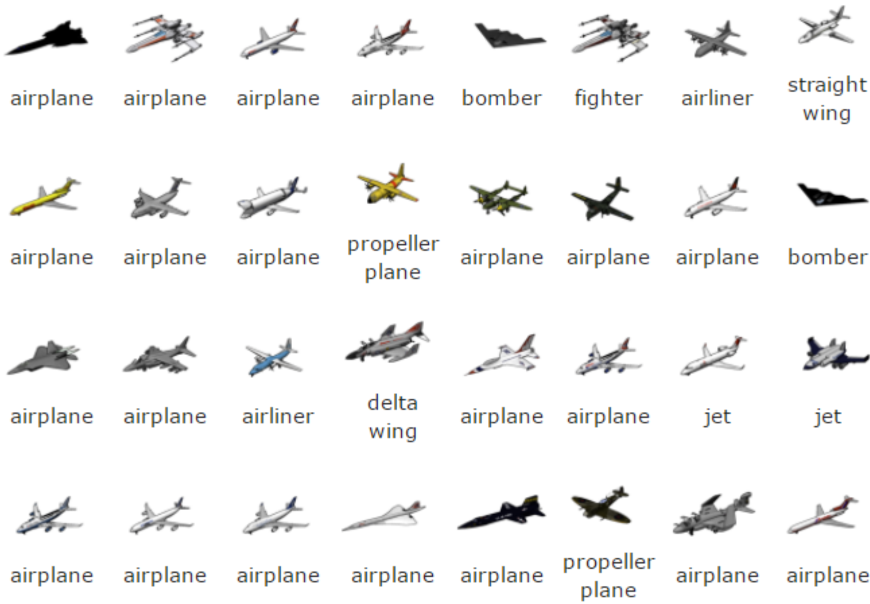
[Qi et al. 16] 3D vs Multi-view

3D Classification

- Training typically on ShapeNet
  - > 55k CAD models

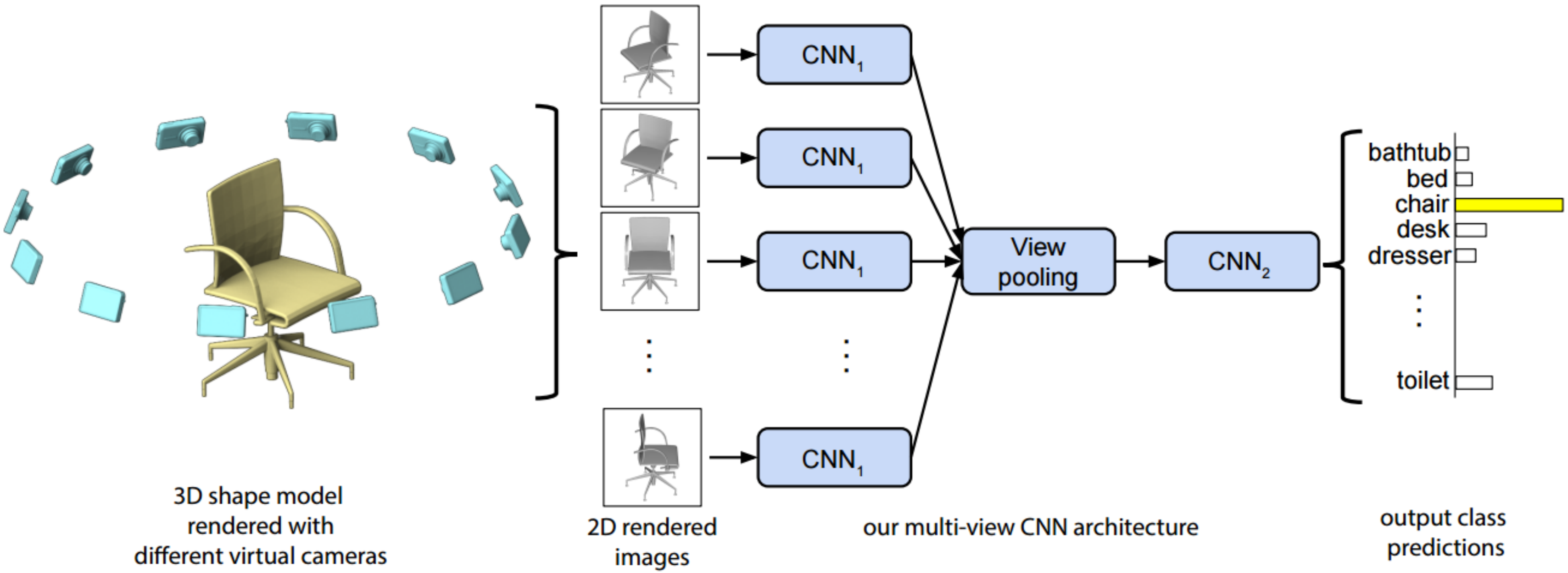
Network	Single-Ori	Multi-Ori
E2E-[30]	83.0	87.8
VoxNet[21]	83.8	85.9
3D-NIN	86.1	88.5
Ours-SubvolumeSup	**87.2**	89.2
Ours-AniProbing	84.4	**89.9**

Table 2. Comparison of performance of volumetric CNN architectures. Numbers reported are classification accuracy on ModelNet40. Results from E2E-[30] (end-to-end learning version) and VoxNet [21] are obtained by ourselves. All experiments are using the same set of azimuth and elevation augmented data.



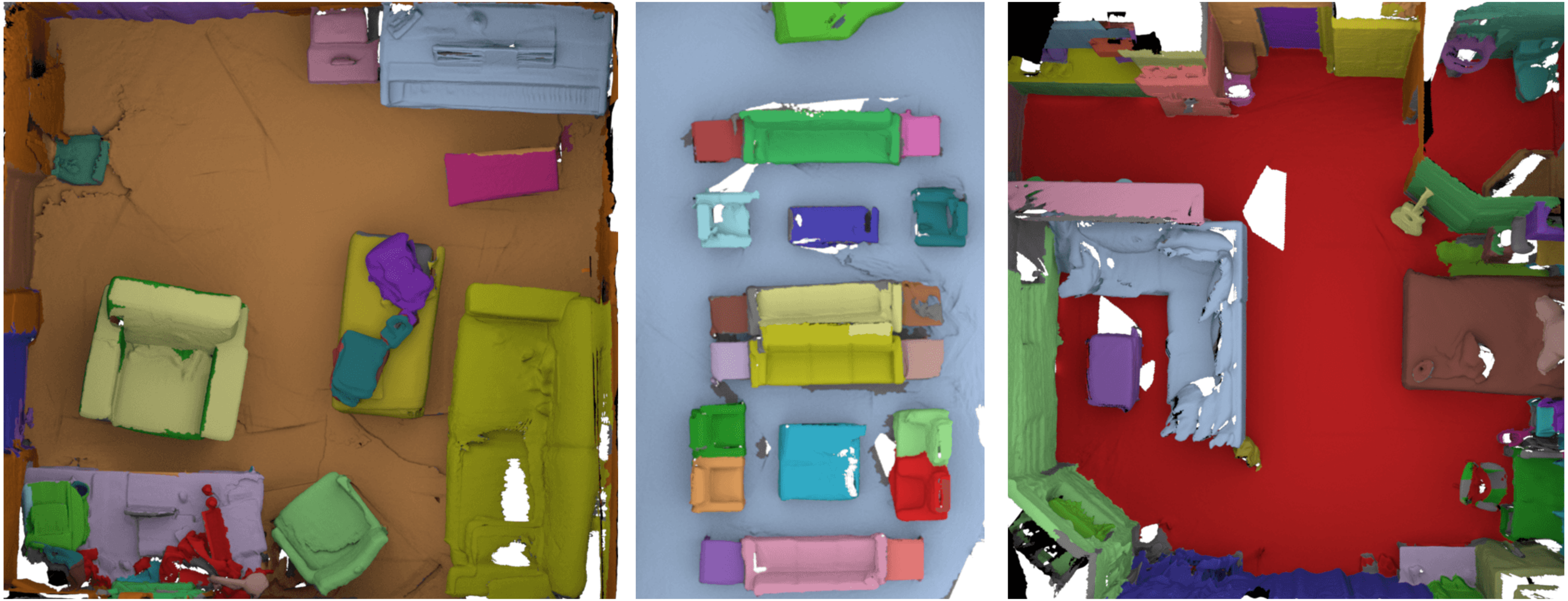
[Chang et al. 15] ShapeNet -- [Qi et al. 16] 3D vs Multi-view

Multi-View CNNs (aggregate 2D)



[Su et al. 15] Multi-view CNN

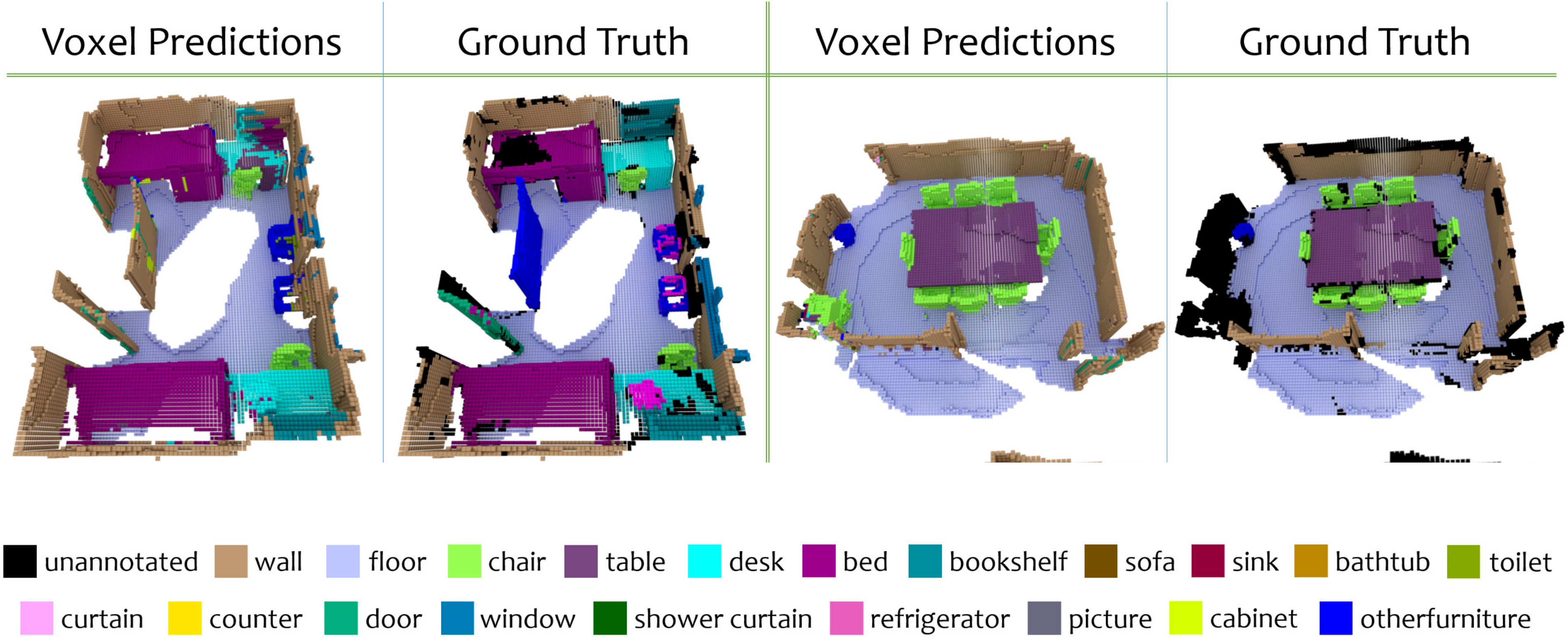
3D CNNs on Real-World Data



1500 densely annotated 3D scans; 2.5 mio RGB-D frames

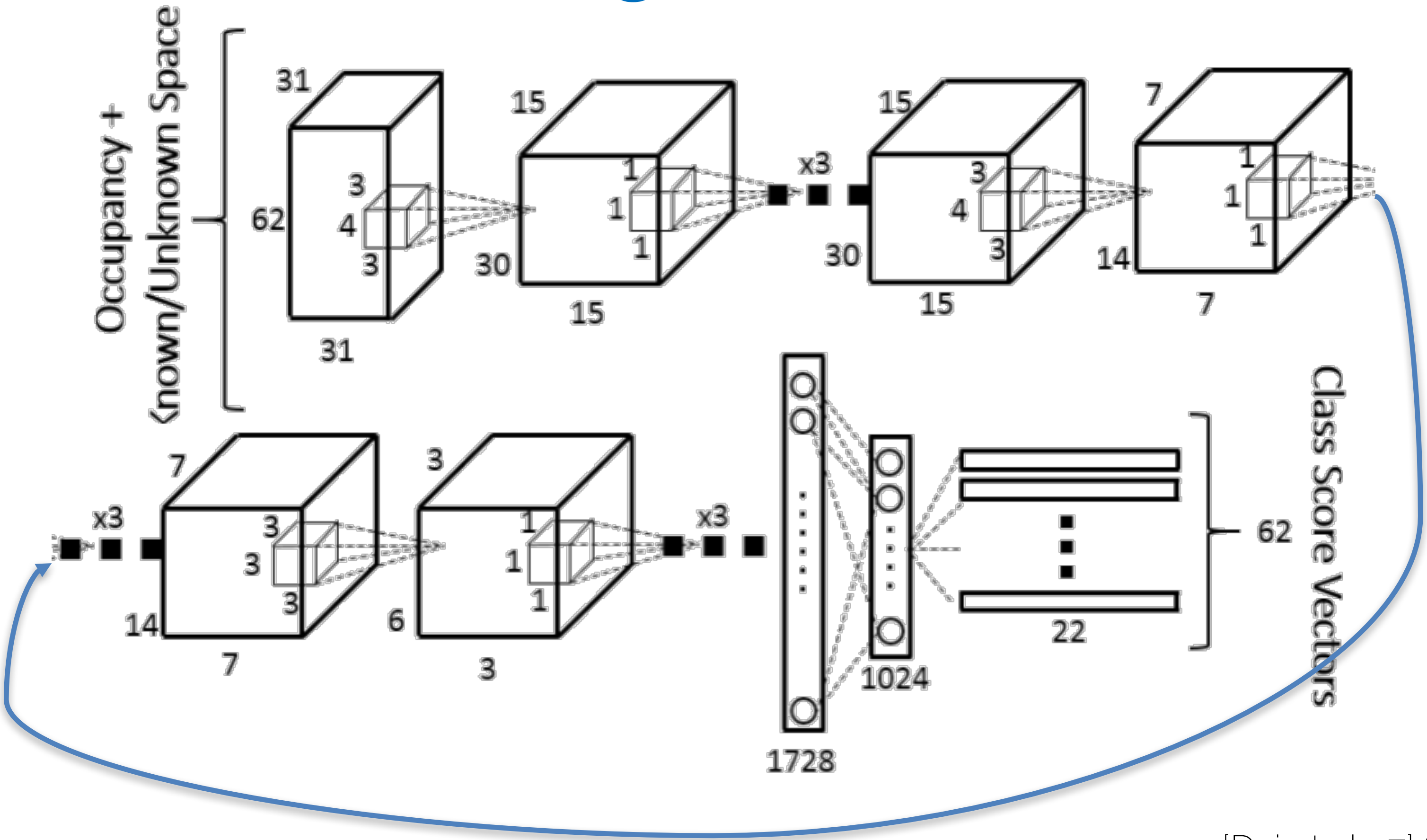
[Dai et al. 17] ScanNet

Semantic Segmentation in 3D



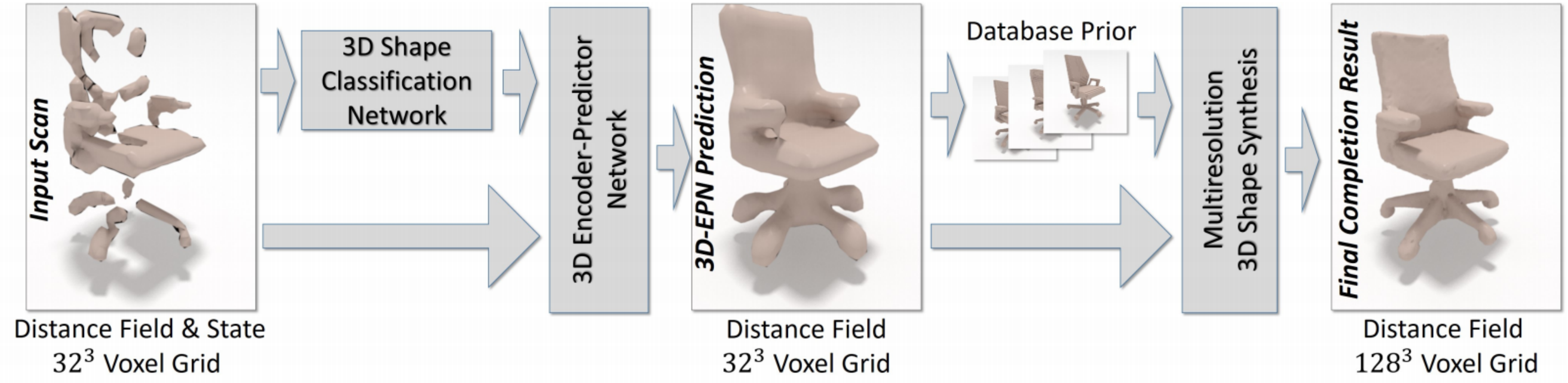
[Dai et al. 17] ScanNet

Semantic Segmentation in 3D



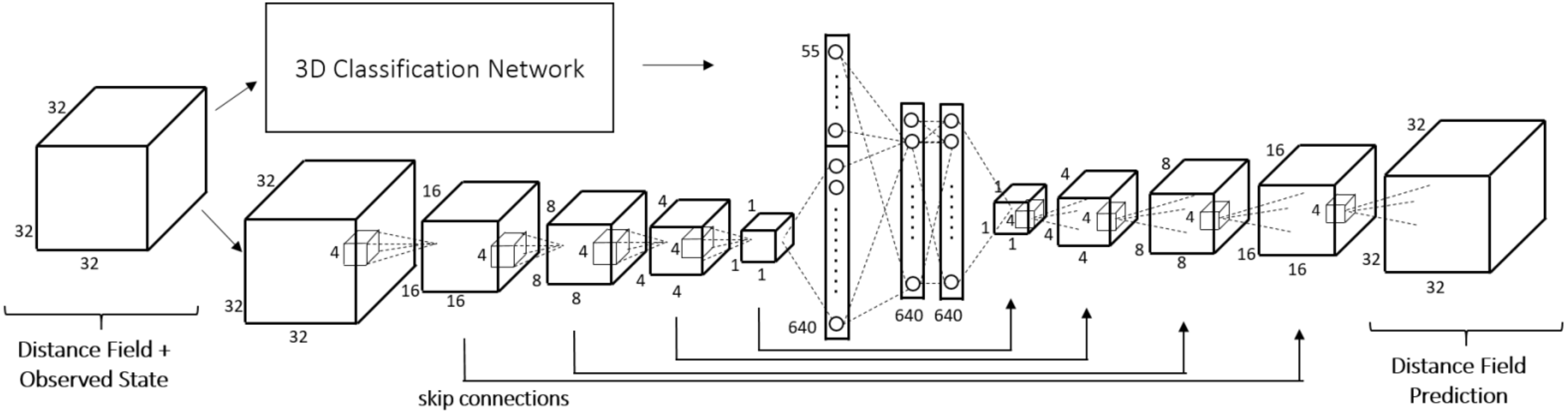
[Dai et al. 17] ScanNet

3D Shape Completion



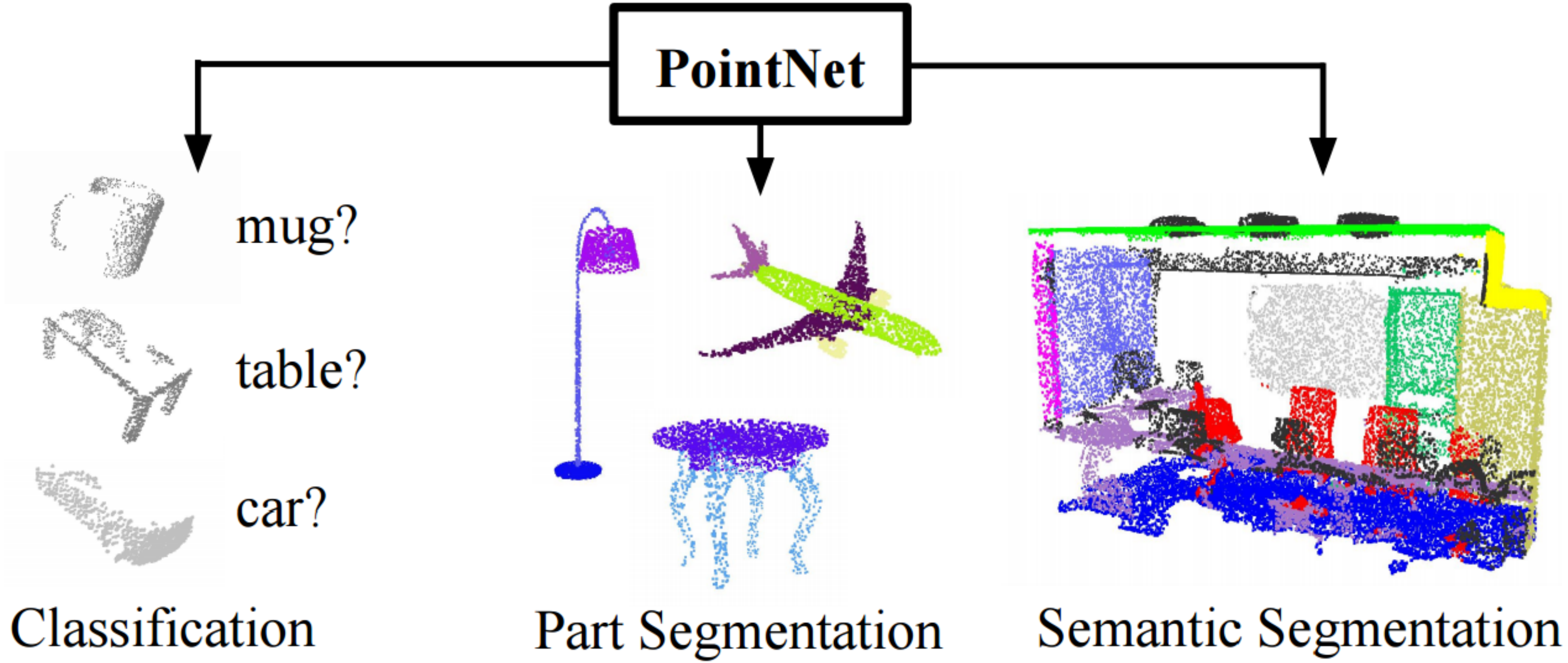
[Dai et al. 17] CNNComplete

3D Shape Completion



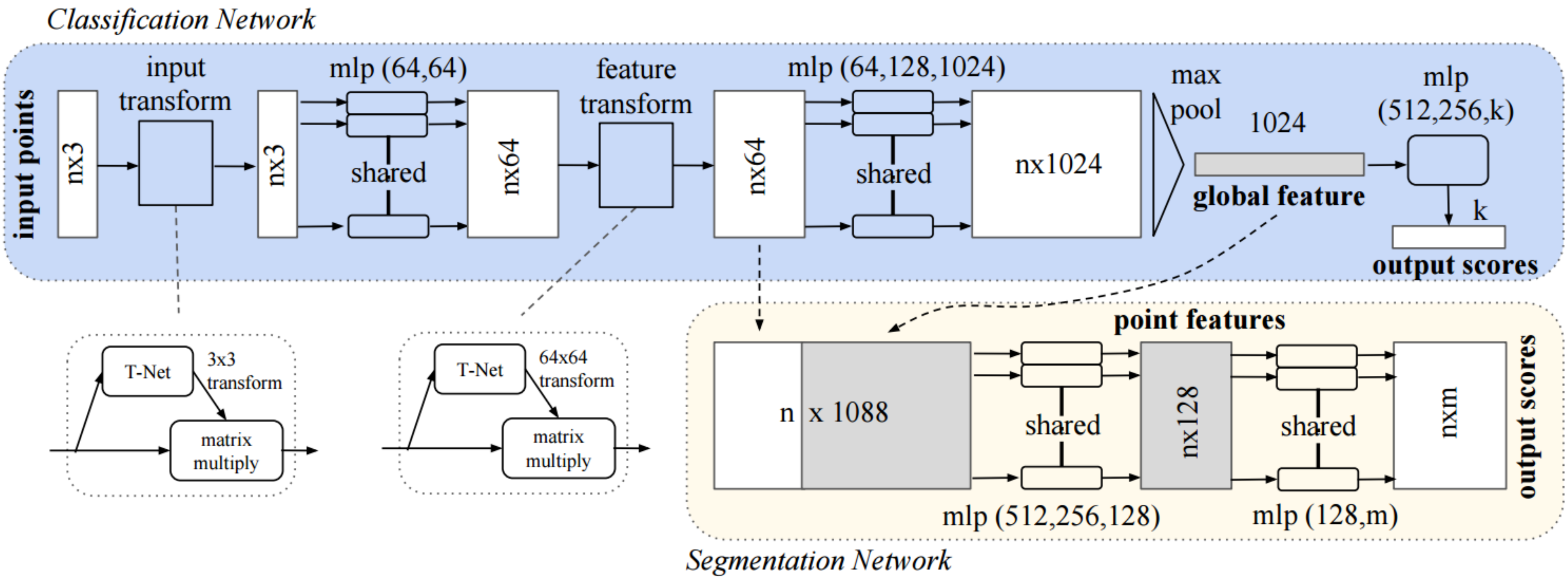
[Dai et al. 17] CNNComplete

Convolutions on Point Clouds: PointNet



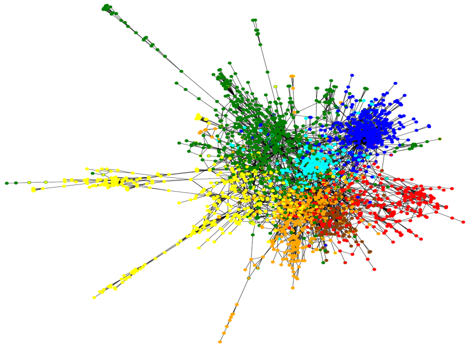
[Qi et al. 17] PointNet

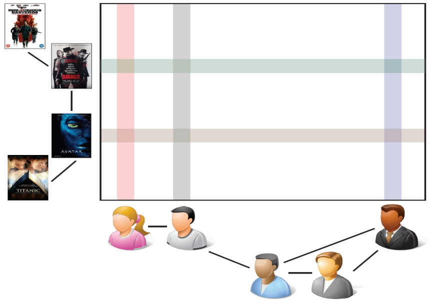
Convolutions on Point Clouds: PointNet

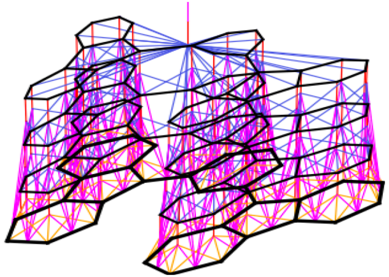


[Qi et al. 17] PointNet

CNNs on Meshes and Graphs



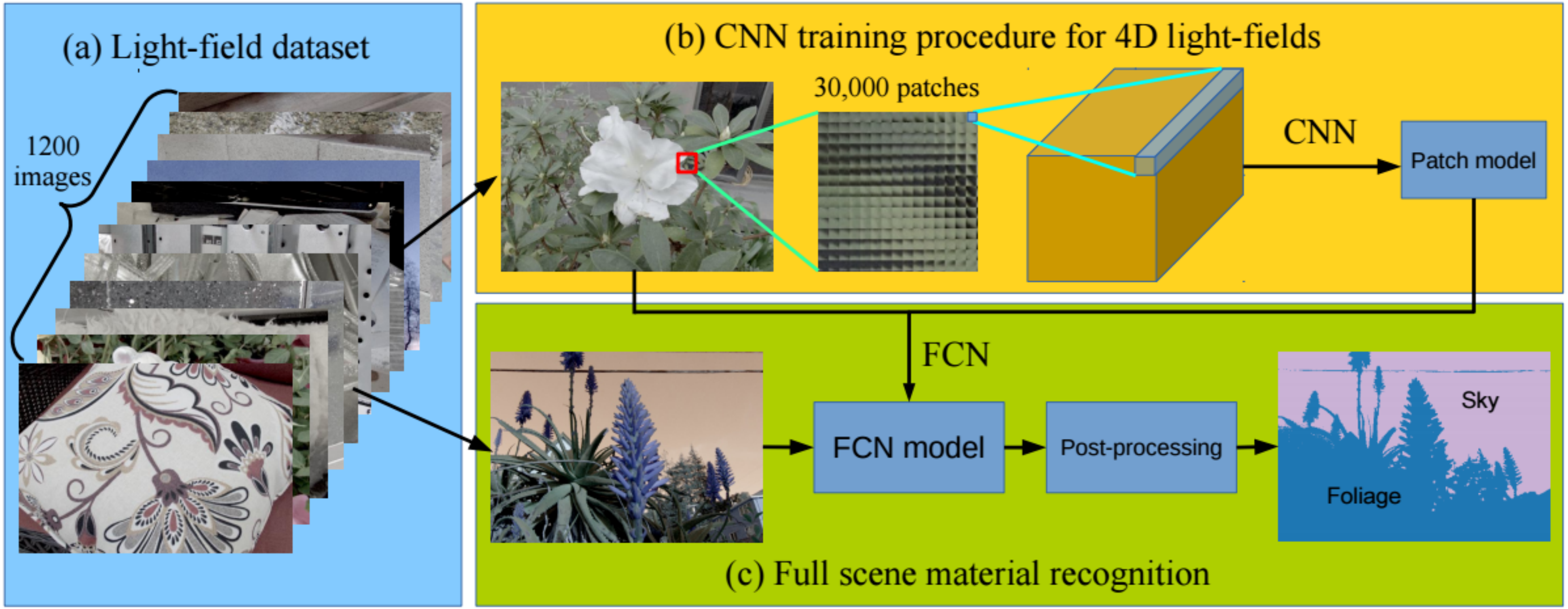




Graphs, Manifolds, etc..

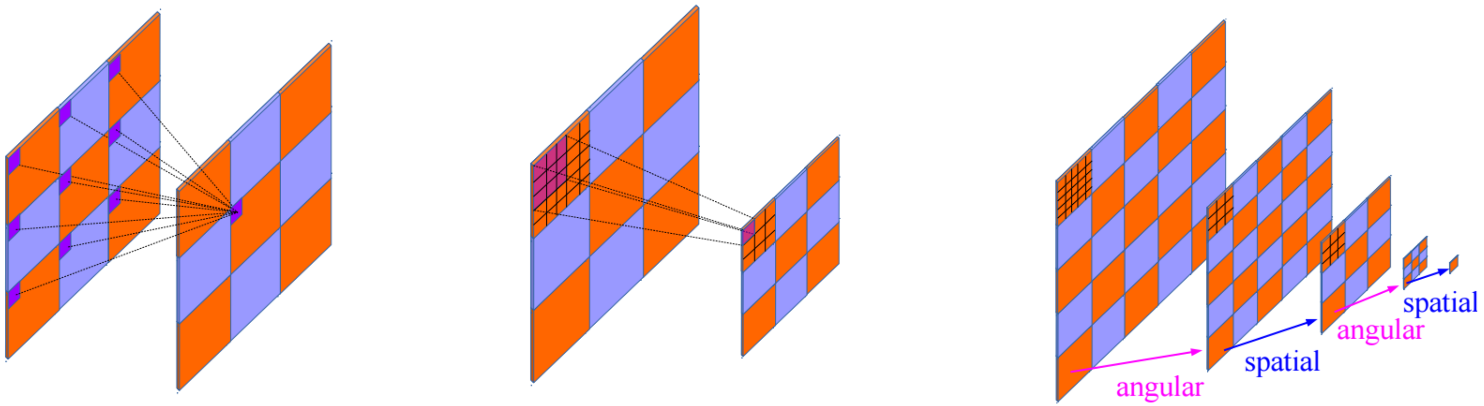
-> See Michael Bronstein's course last week/tomorrow!

4D CNNs ???



[Wang et al. 16] 4D Light-Fields

4D CNNs ???



(a) spatial filter (b) angular filter (c) interleaved filter

Fig. 5: *(a)(b) New spatial and angular filters on a remap light-field image. The pooling feature is also implemented in a similar way. (c) By interleaving the angular and spatial filters (or vice versa), we mimic the structure of a 4D filter.*

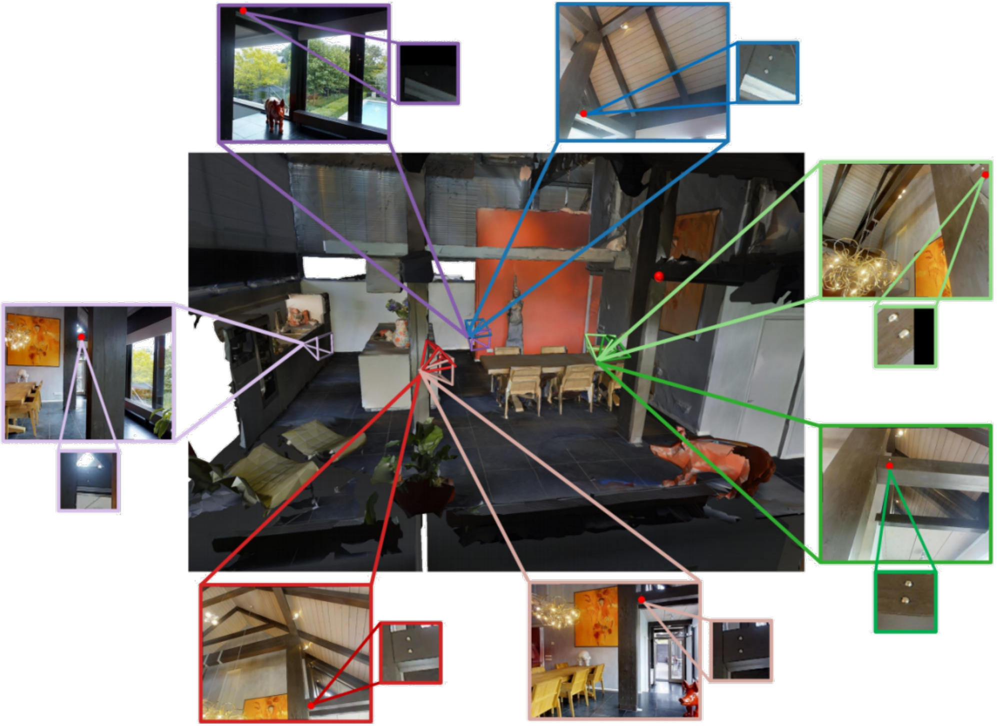
[Wang et al. 16] 4D Light-Fields

Self-Supervised Learning

- Supervised vs Self-supervised
- Weakly-supervised vs Self-supervised
  - Good labeled data is *always* an issue

Self-Supervised Learning

- E.g., learning to match Key Points via 3D



SURF	46.8%
SIFT	37.8%
ResNet-50 w/ Matterport3D	10.6%
ResNet-50 w/ SUN3D	10.5%
ResNet-50 w/ Matterport3D + SUN3D	9.2%

Error (%) at 95% recall tested on SUN3D

Self-Supervised Learning

- Feature matching
- Normal predictions
- Novel view prediction
- Camera pose between two images
- Depth map prediction / in-painting depth
- Optical flow / Scene flow
- Generate color for depth geometry
- …

*Always think if there are *free* training labels!!!*

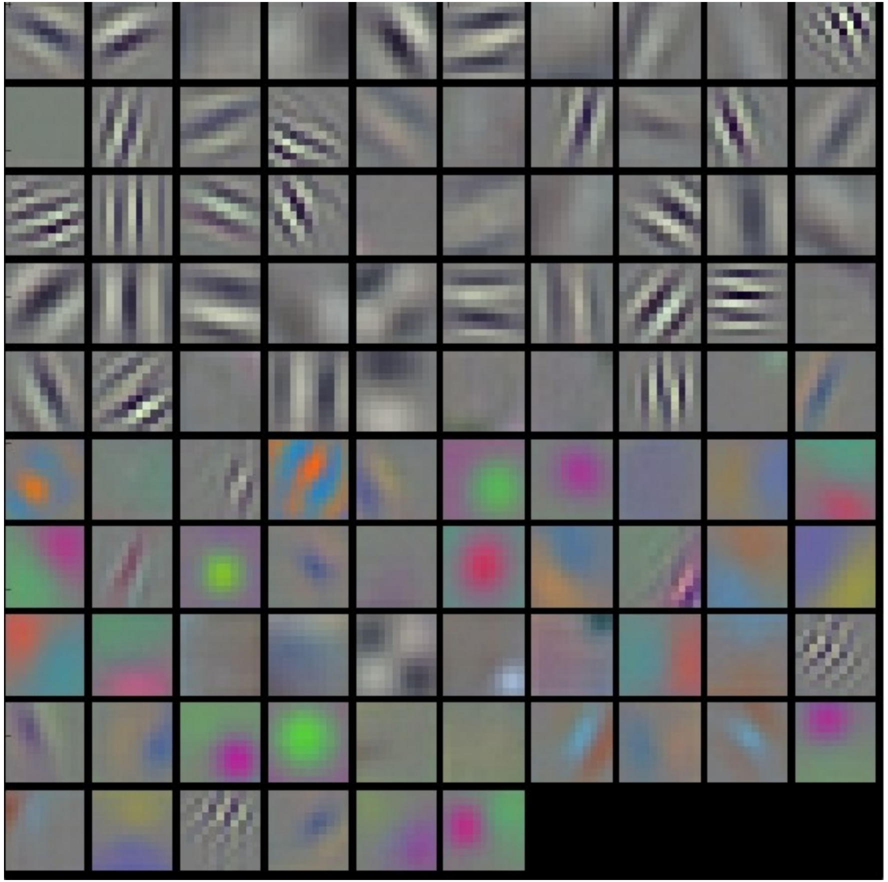
Visualization of ConvNets

Visualization of ConvNets

- Visualization of Features
- Visualization of Activations
- Visualization of Gradients
- T-SNE Visualization
- DeepDream
- ….

Visualization is a great way for debugging!

Visualization of Features



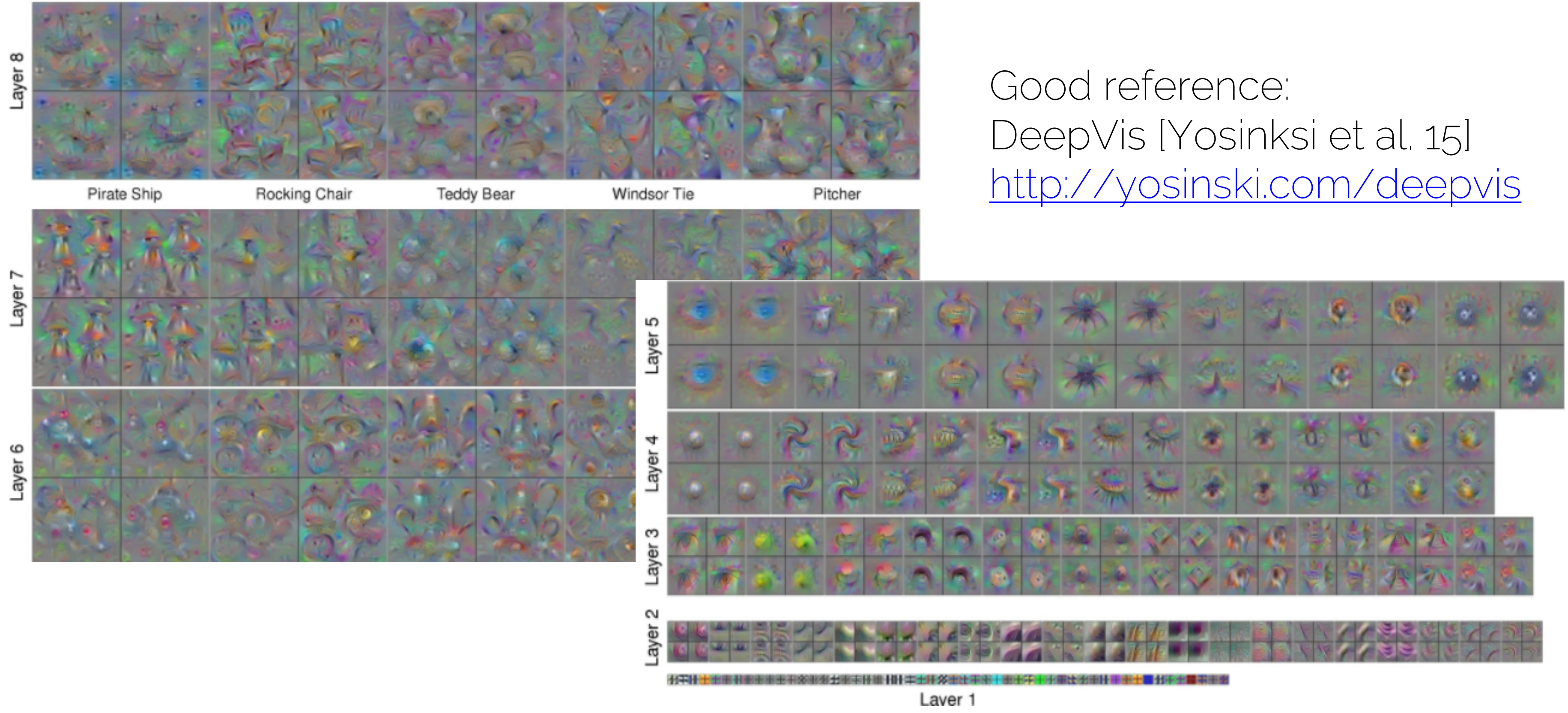
Visualization of AlexNet Features
first Conv Layer (weights visualized)

Color clusters are due to AlexNet streams

Other layers are not so easy to visualize typically need projection first

Credit: Li/Karpathy/Johnson

Visualization of Gradients



Good reference:
DeepVis [Yosinksi et al. 15]
http://yosinski.com/deepvis

Deep Visualization Toolbox

yosinski.com/deepvis

#deepvis


Jason Yosinski


Jeff Clune


Anh Nguyen


Thomas Fuchs


Hod Lipson

Cornell University

University of Wyoming

NASA Jet Propulsion Laboratory
California Institute of Technology

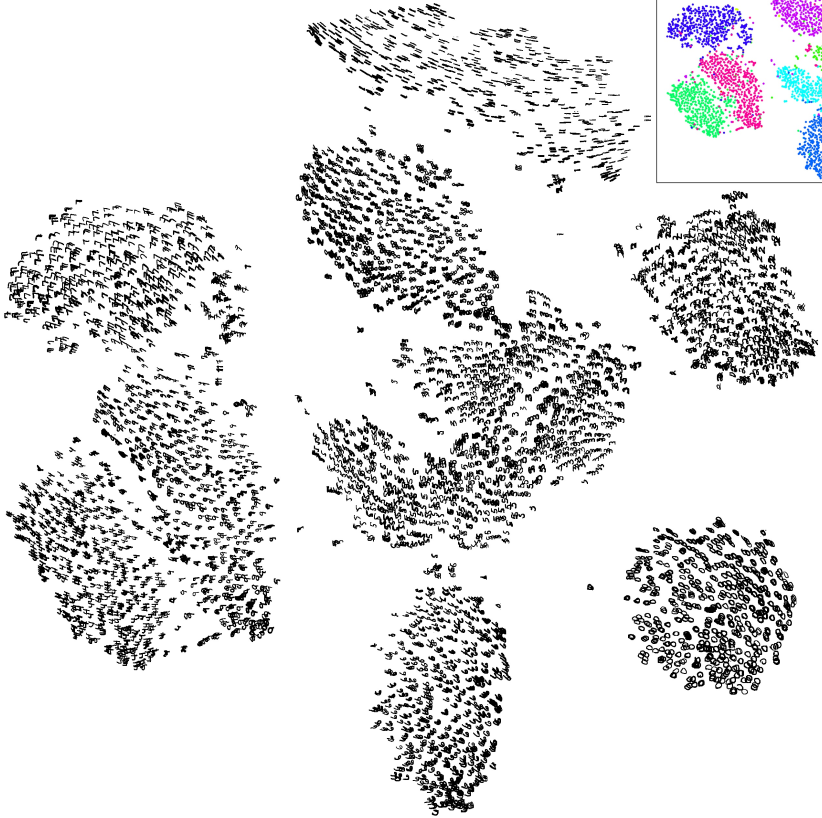
t-SNE Visualization

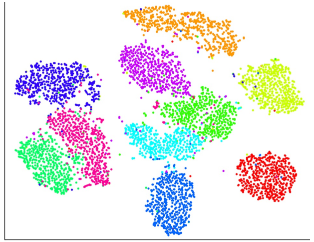
t-Distributed Stochastic Neighbor Embedding (t-SNE)

- Map high-dimensional embedding to 2D map
- Add samples from dataset according to their features to large image
- Very useful to spot clusters and debug embedding

[van der Maaten et al.] t-SNE

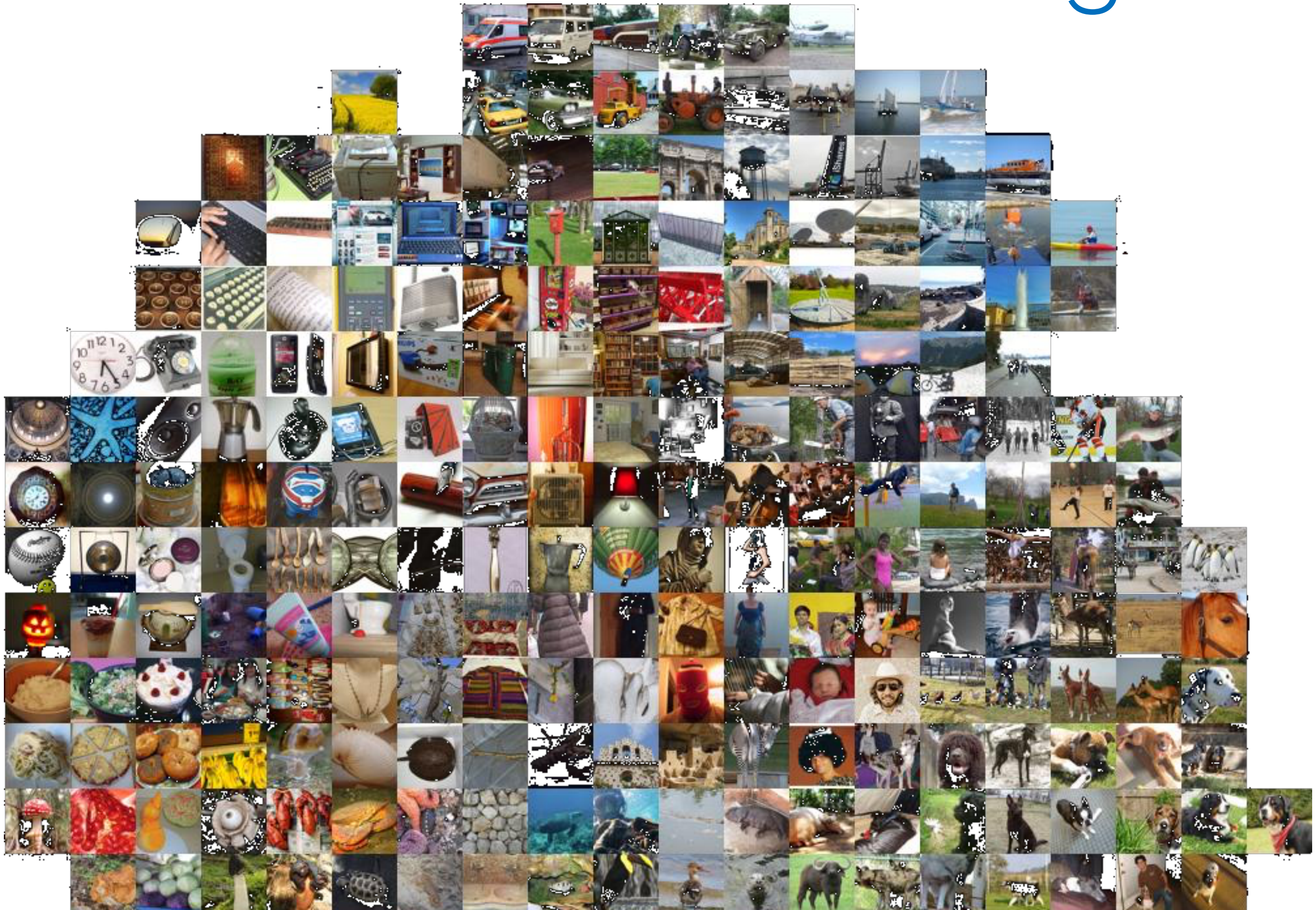
t-SNE Visualization: MNIST





[van der Maaten et al.] t-SNE

t-SNE Visualization: ImageNet



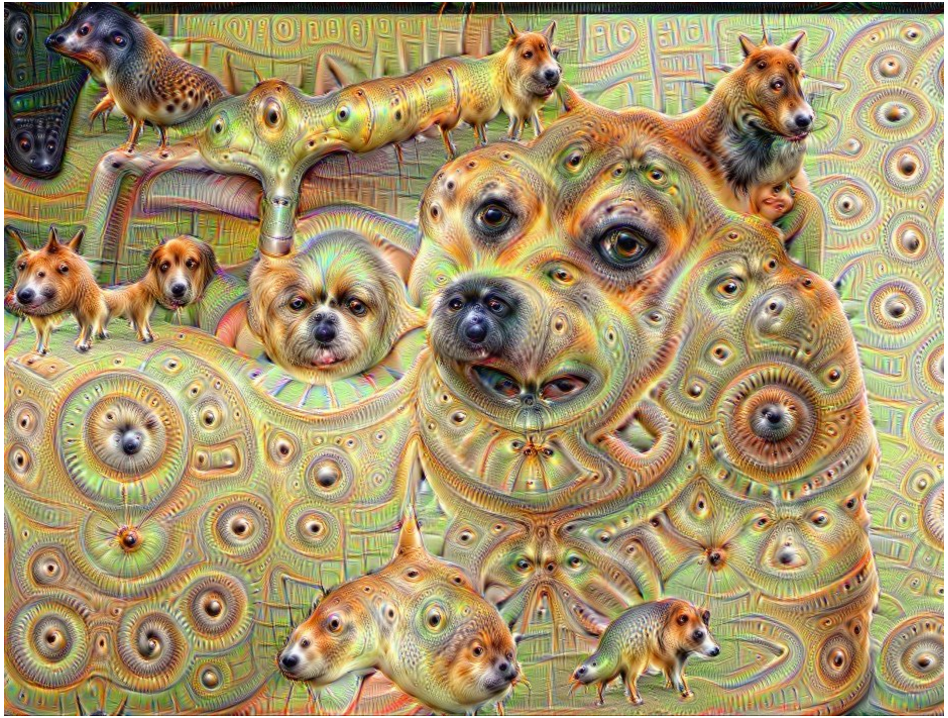
Karpathy

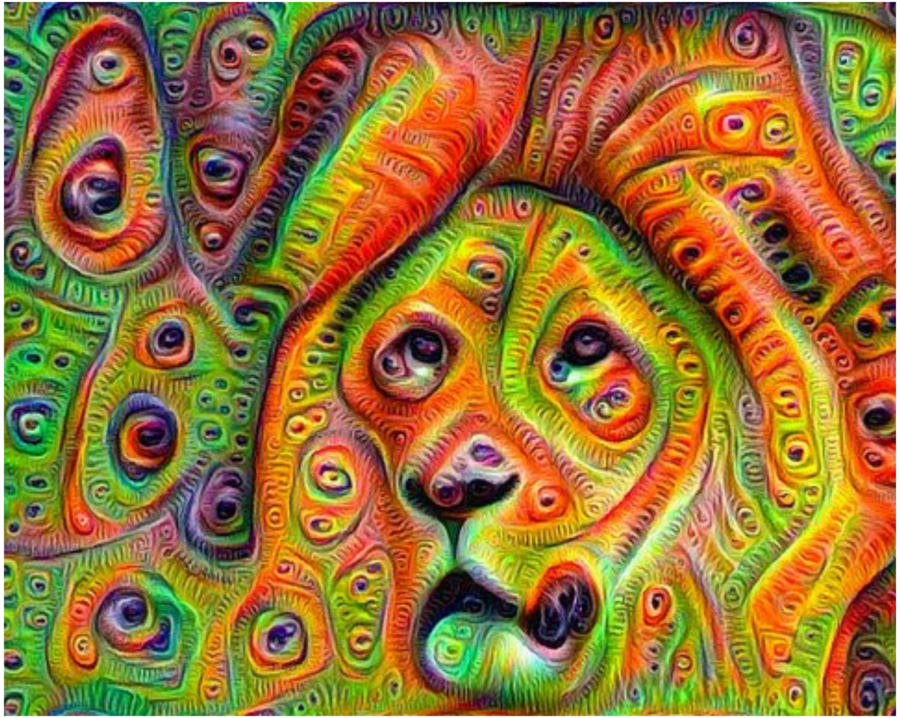
t-SNE Visualization: ShapeNet

DeepDream

[Mordvintsev et al. 15] DeepDream

DeepDream





[Mordvintsev et al. 15] DeepDream

CNNs in Practice

- Hints for projects:
  - Don't train from scratch
  - Use transfer learning when possible
  - Think about smart ways for data augmentation
  - Pre-train with auto-encoder if only small labeled dataset
  - Check training progress early on!

CNNs in Practice

- Hints for projects:
  - Start simple! E.g., first overfit to a single training sample
  - Always try simple architectures first
    - ResNet is not always a good start; VGG typically easier
  - Estimate timings (how long for an epoch?)
  - Double check implementation (is data read correctly?)

Administrative Things

- Thursday July 13th: Vis cont'd, RNN, LSTM!
- Tomorrow:
  - 2nd part of Michael Bronstein "Geometric Deep Learning" course

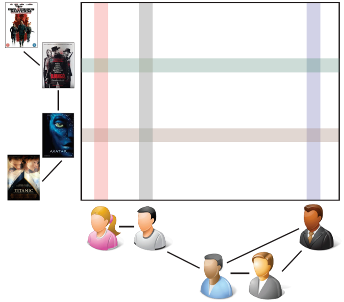
Special Course:

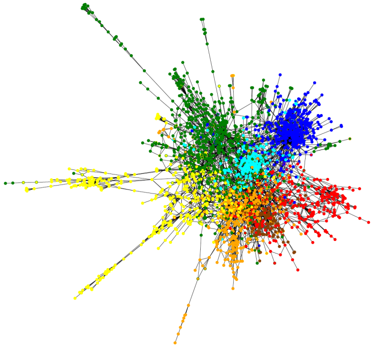
Geometric deep learning on graphs and manifolds

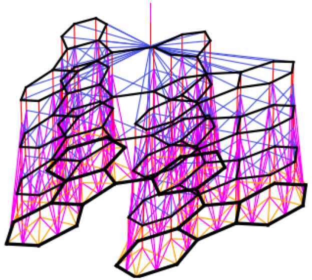
Going beyond Euclidean data

Michael Bronstein

USI Lugano / Tel Aviv University / Intel Perceptual Computing / TUM IAS







Preliminary: scheduled for Fri 30/6 and 7/7 (2pm to 4pm)

-> in our tutorial room