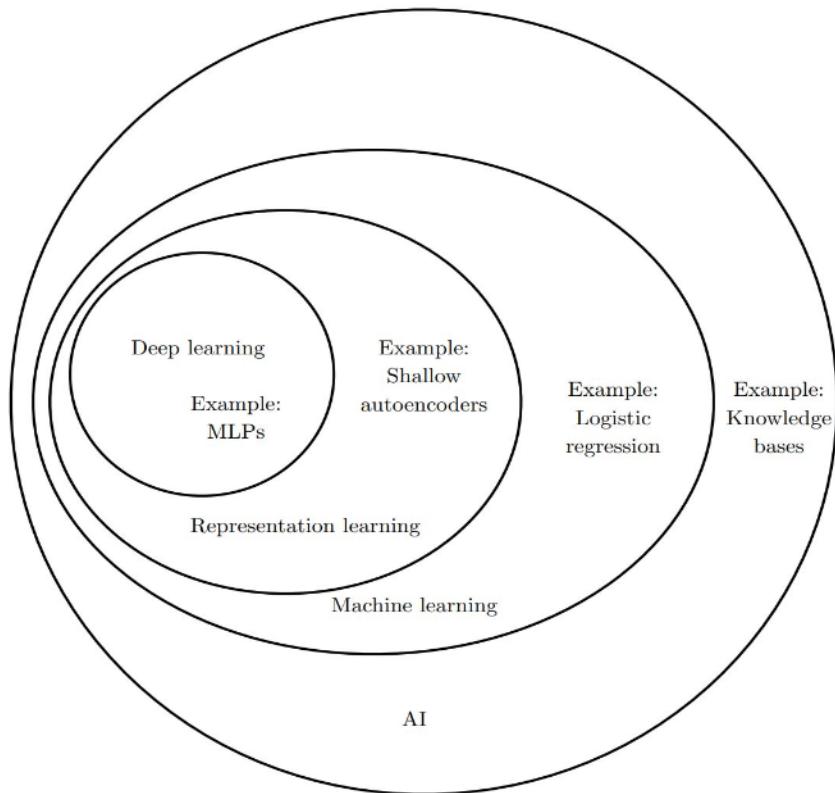


Deep Learning is one way to learn features



Deep Unsupervised Learning:

1. Learn representations without labels
2. Subset of Deep Learning, which is a subset of Representation Learning, which is a subset of Machine Learning

Self-Supervised Learning

1. Often used interchangeably with unsupervised learning
2. Self-Supervised: Create your own supervision through pretext tasks

Why Self-Supervised Learning?

- Expense of producing a new dataset for each task
 - Prepare labeling manuals, categories, hiring humans, creating GUIs, storage pipelines, etc.
- Good supervision may not be cheap (ex: medicine, legal)
- Take advantage of vast amount of unlabeled data on the internet (images, videos, language).
- Cognitive motivation: How animals / babies learn

What is Self-Supervised Learning?

- A version of unsupervised learning where data provides the supervision.
- In general, withhold some part of the data and the task a neural network to predict it from the remaining parts.
- Details decide what proxy loss or pretext task the network tries to solve, and depending on the quality of the task, good semantic features can be obtained without actual labels.

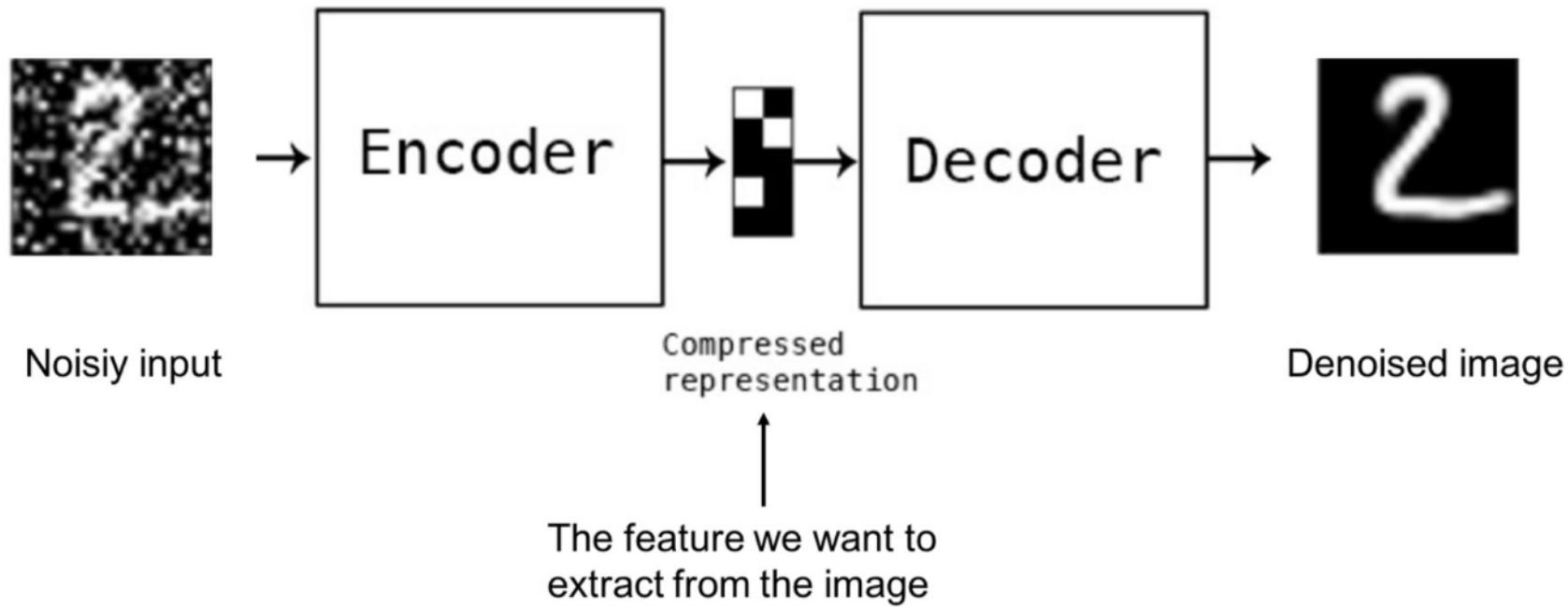
Motivation

- Supervised learning success story is heavily because of the utility of pre-trained classifier features for commercially useful downstream tasks like segmentation, detection, etc.
- Recipe is clear: Collect a large labeled dataset, train a model, deploy. Good data and sufficient data are what you need.
- Goal of self-supervised learning:
 - Learn equally good (if not better) features without supervision
 - Be able to deploy similar quality systems without relying on too many labels for the downstream tasks
 - Generalize better potentially because you learn more about the world

Cognitive Principles

- Reconstruct from a corrupted (or partial) version
 - Denoising Autoencoder
 - In-painting
 - Colorization, Split-Brain Autoencoder
- Visual common sense tasks
 - Relative patch prediction
 - Jigsaw puzzles
 - Rotation
- Contrastive Learning
 - word2vec
 - Contrastive Predictive Coding (CPC)
 - Instance Discrimination
 - Recent State-of-the-art progress

Denoising Autoencoder

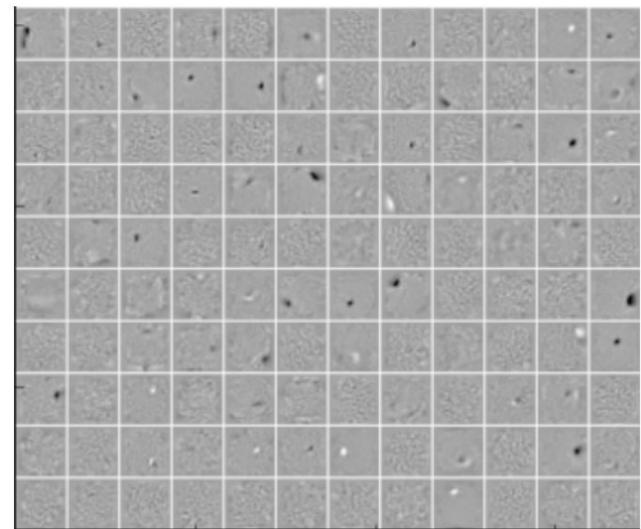


Denoising Autoencoder

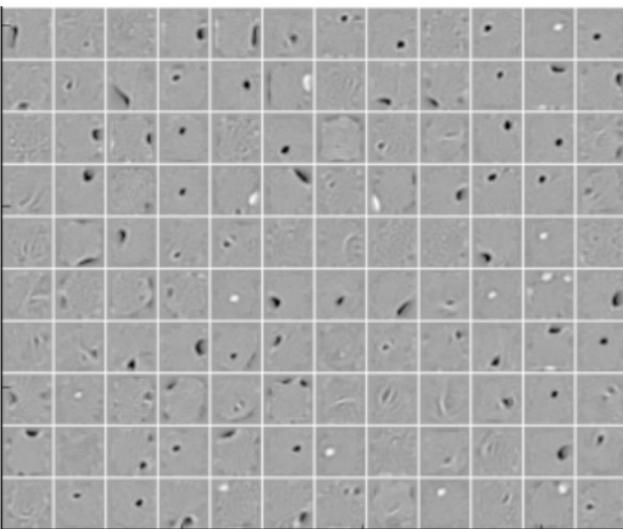
- Additive isotropic *Gaussian noise* (GS): $\tilde{\mathbf{x}}|\mathbf{x} \sim \mathcal{N}(\mathbf{x}, \sigma^2 I)$;
- *Masking noise* (MN): a fraction v of the elements of \mathbf{x} (chosen at random for each example) is forced to 0;
- *Salt-and-pepper noise* (SP): a fraction v of the elements of \mathbf{x} (chosen at random for each example) is set to their minimum or maximum possible value (typically 0 or 1) according to a fair coin flip.

Vincent et al 2010

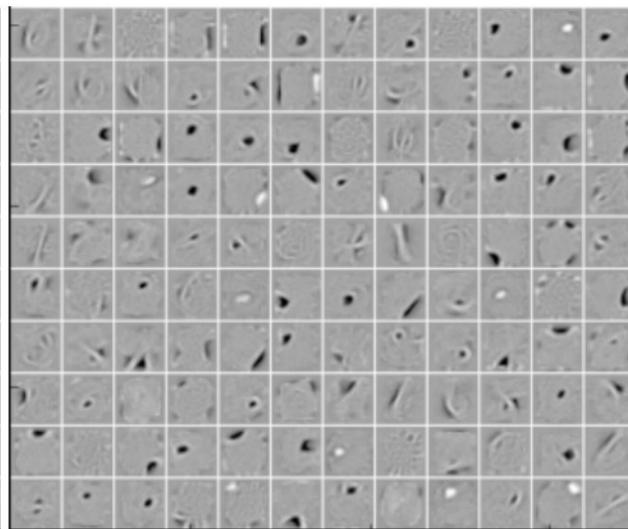
Denoising Autoencoder



(a) No destroyed inputs



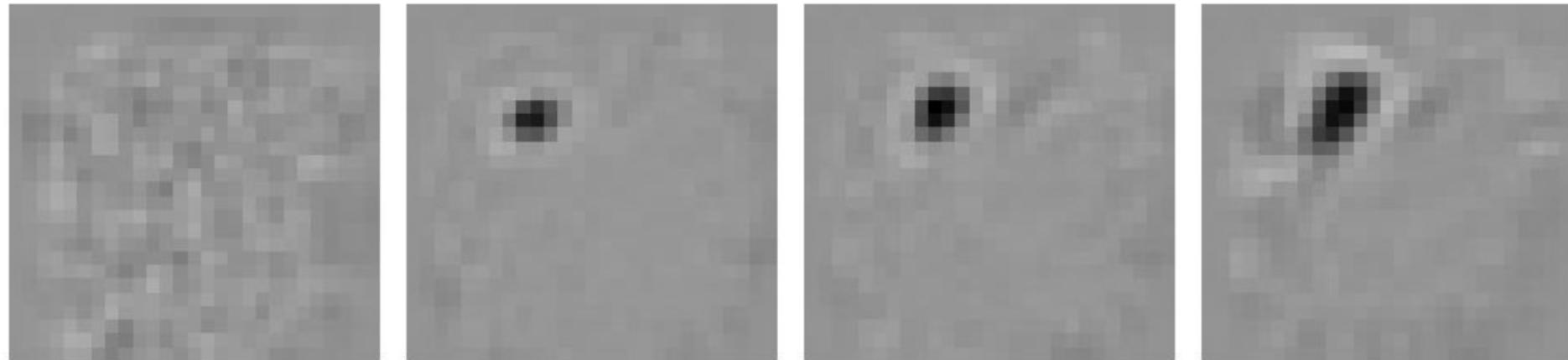
(b) 25% destruction



(c) 50% destruction

Vincent et al 2010

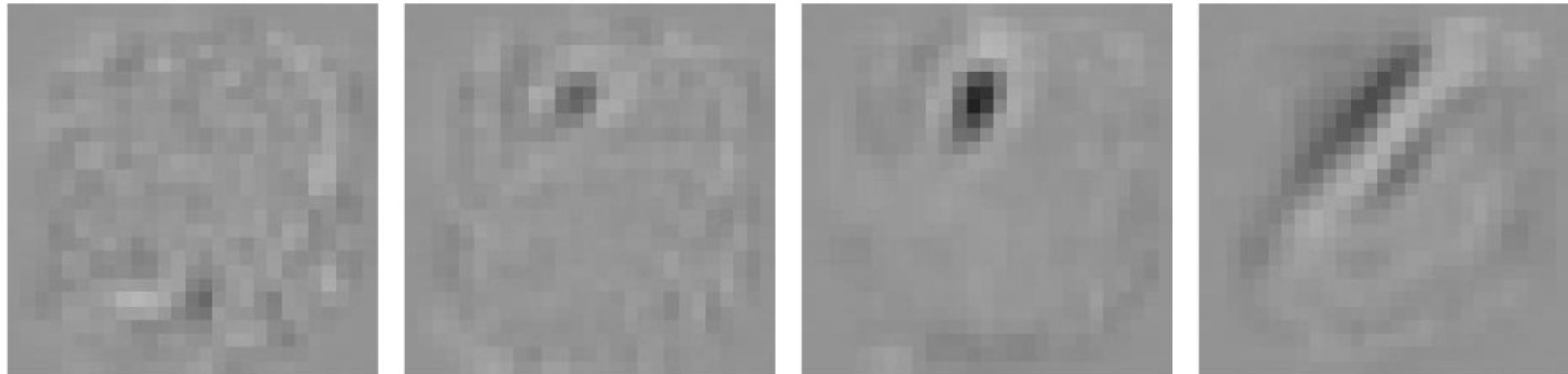
Denoising Autoencoder



(d) Neuron A (0%, 10%, 20%, 50% destruction)

Vincent et al 2010

Denoising Autoencoder



(e) Neuron B (0%, 10%, 20%, 50% destruction)

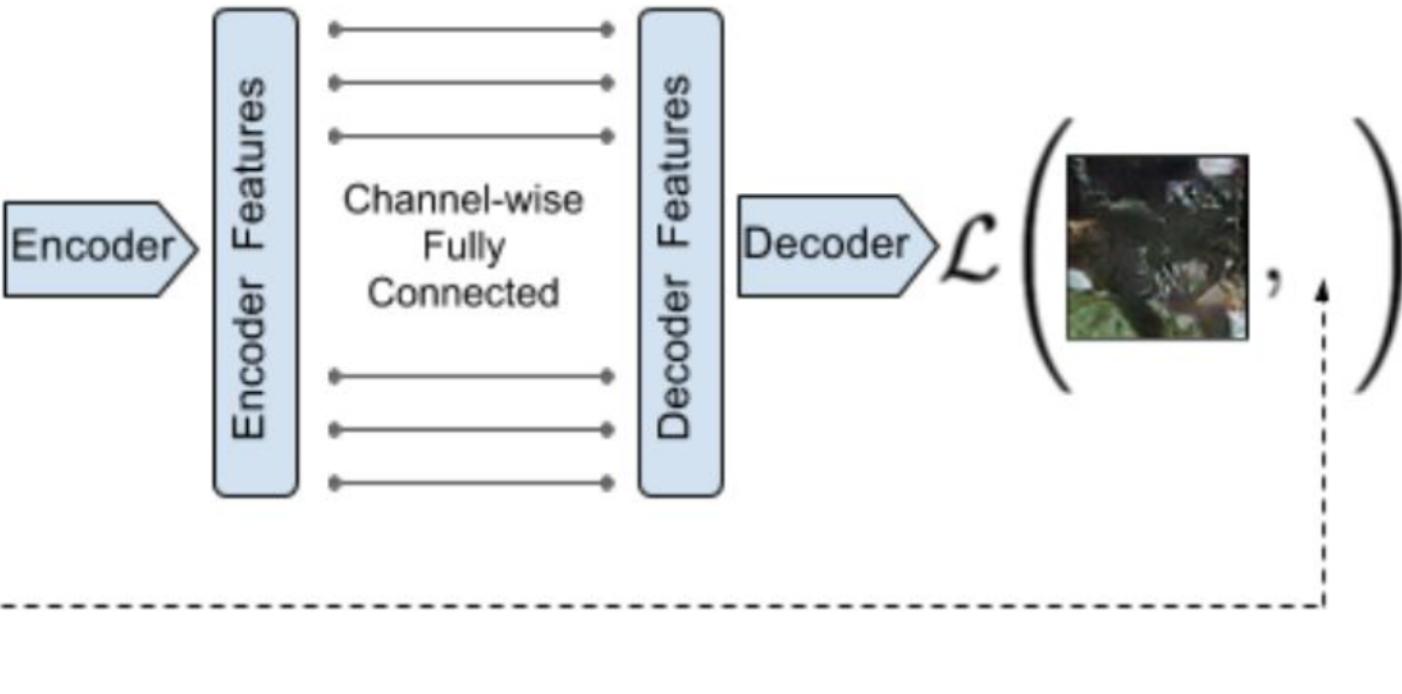
Vincent et al 2010

Predict missing pieces



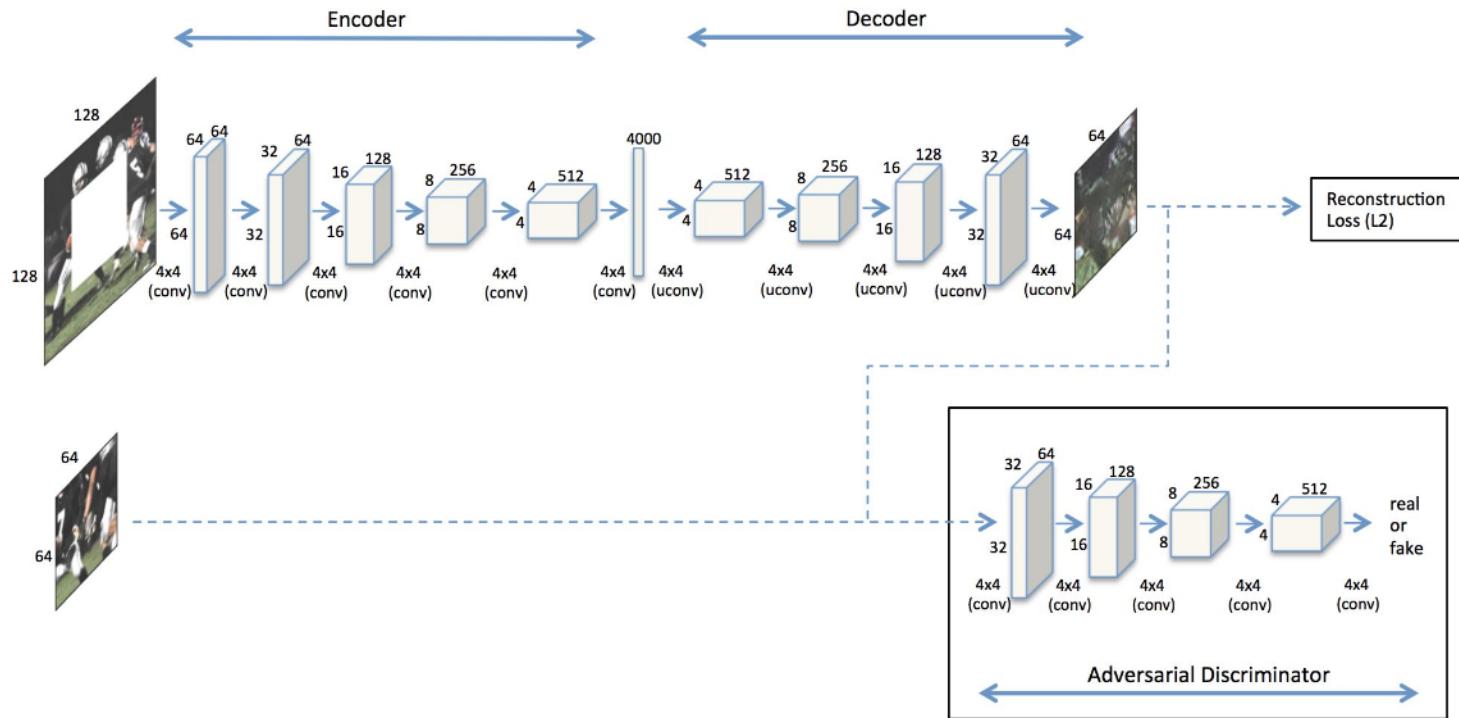
Pathak et al 2016

Context Encoders



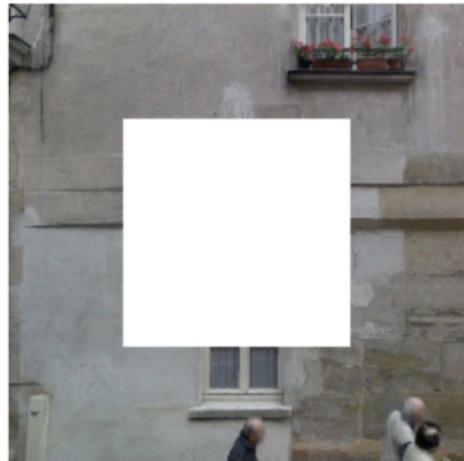
Pathak et al 2016

Context Encoders



Pathak et al 2016

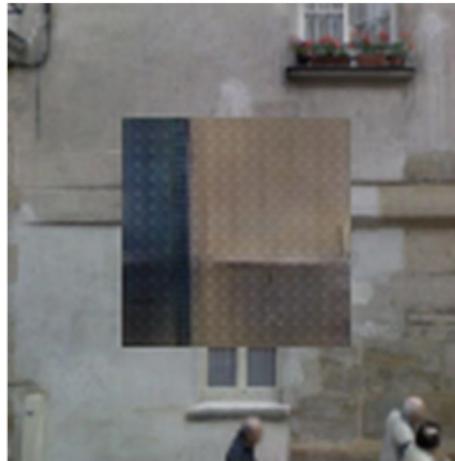
Context Encoders



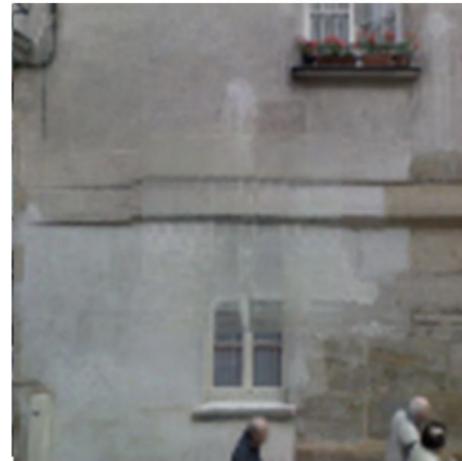
Input Image



L2 Loss



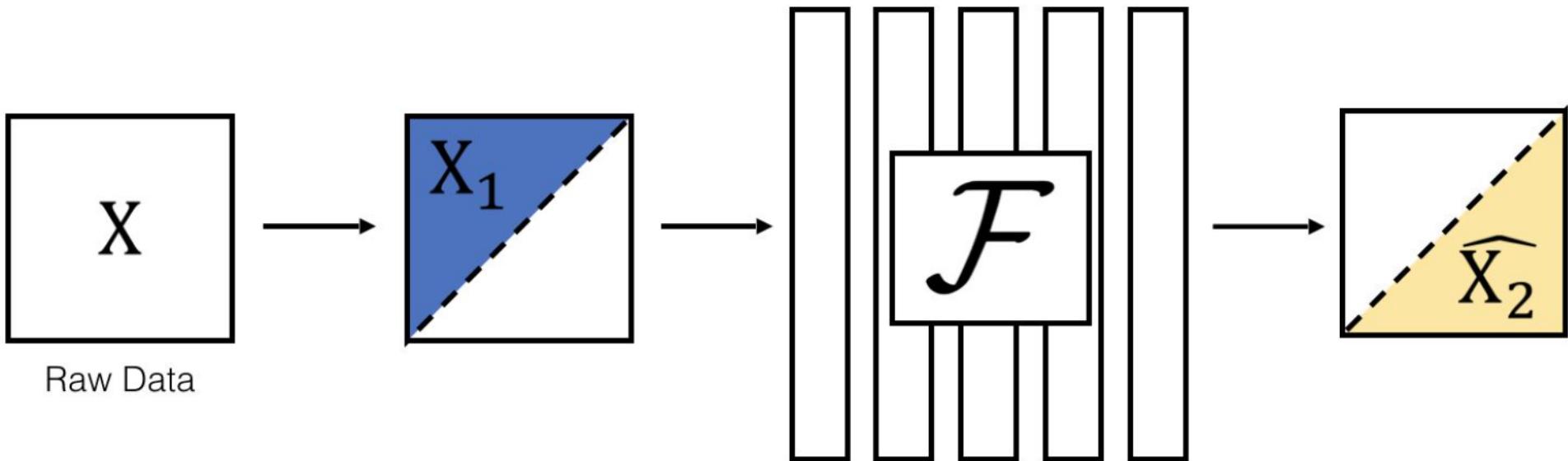
Adversarial Loss



Joint Loss

Pathak et al 2016

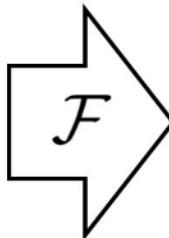
Predicting one view from another



Cross-Channel Encoder

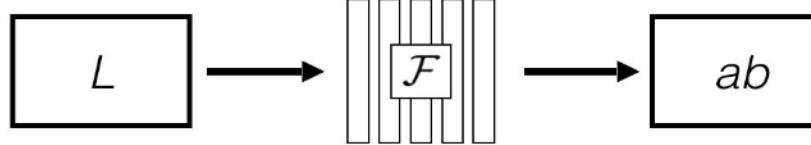
Slide: Richard Zhang

Predicting one view from another



Grayscale image: L channel

$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$



Color information: ab channels

$$\hat{\mathbf{Y}} \in \mathbb{R}^{H \times W \times 2}$$

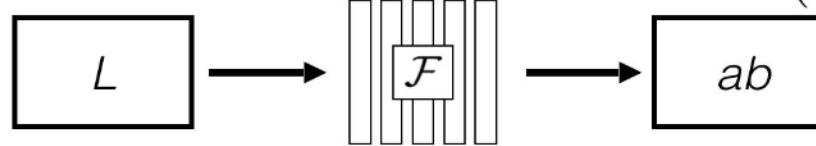
Slide: Richard Zhang

Predicting one view from another



Grayscale image: L channel

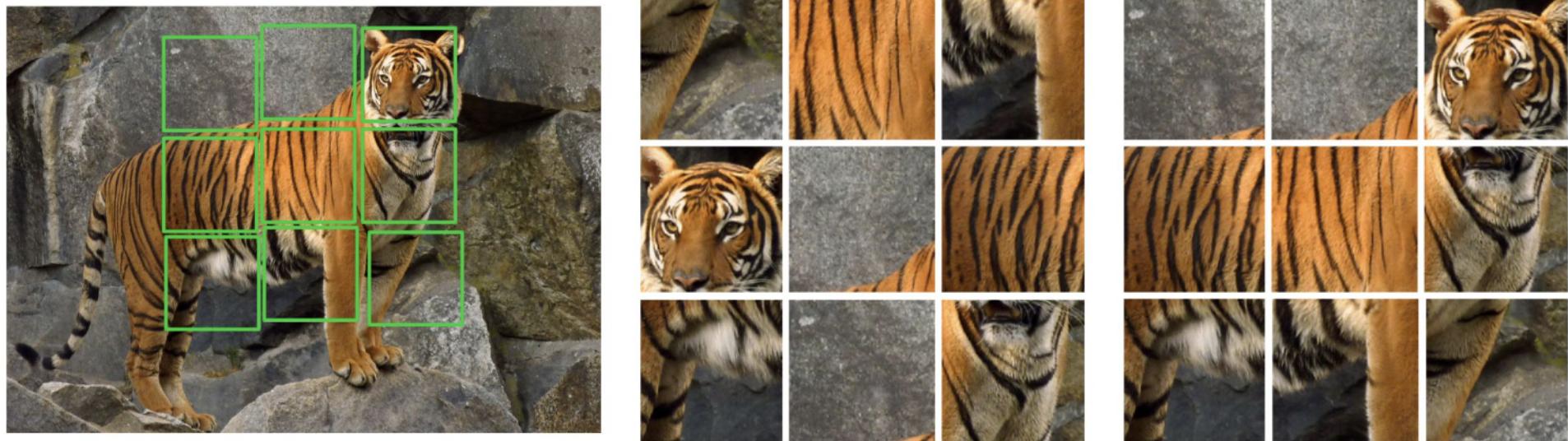
$$\mathbf{X} \in \mathbb{R}^{H \times W \times 1}$$



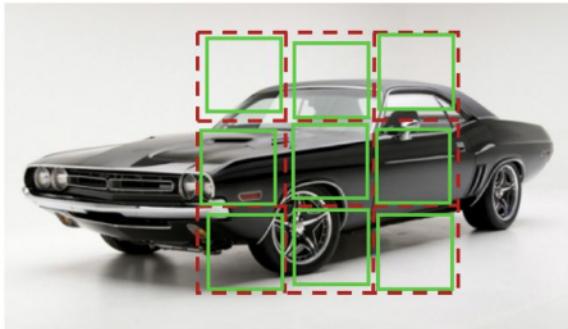
Concatenate (L, ab) channels
 $(\mathbf{X}, \widehat{\mathbf{Y}})$

Slide: Richard Zhang

Solving Jigsaw Puzzles



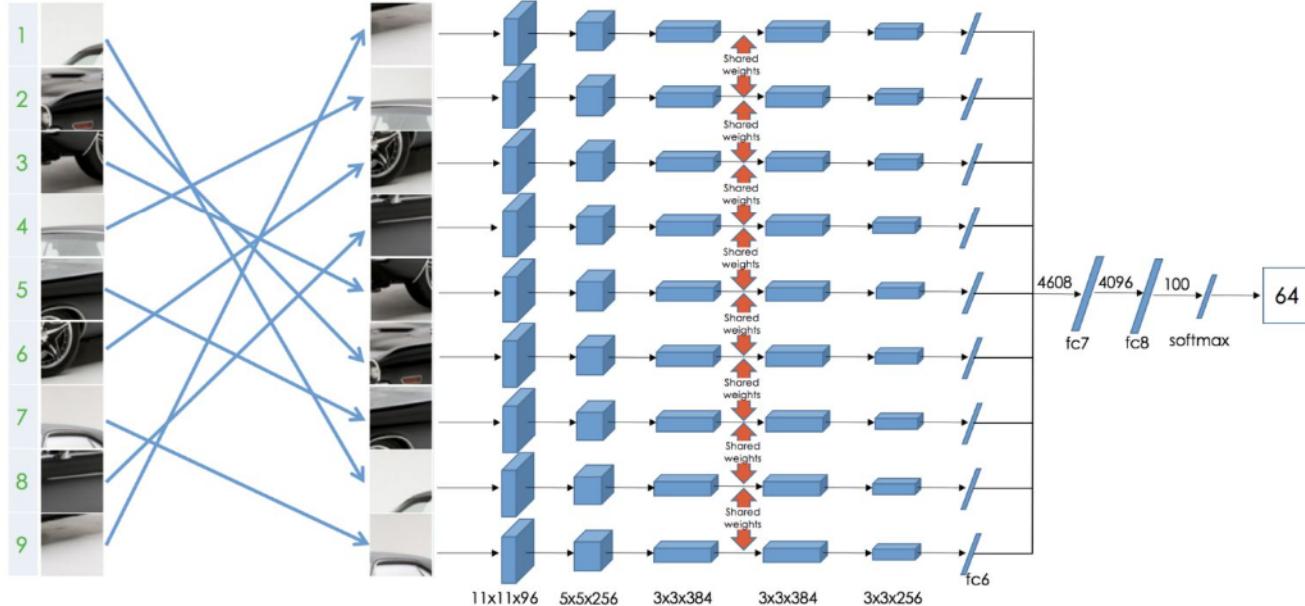
Solving Jigsaw Puzzles



Permutation Set

index	permutation
64	9,4,6,8,3,2,5,1,7

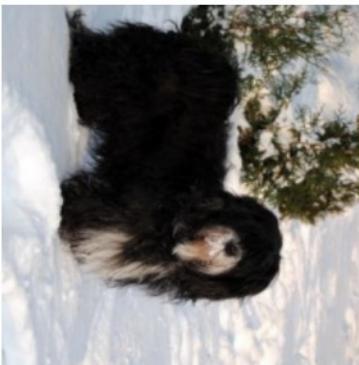
Reorder patches according to the selected permutation



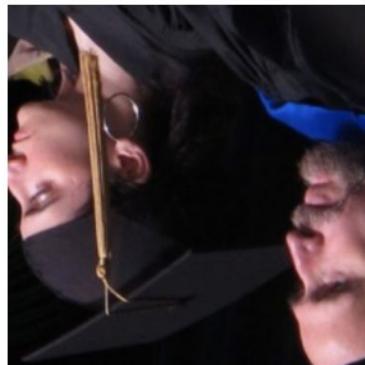
Rotation



90° rotation



270° rotation



180° rotation

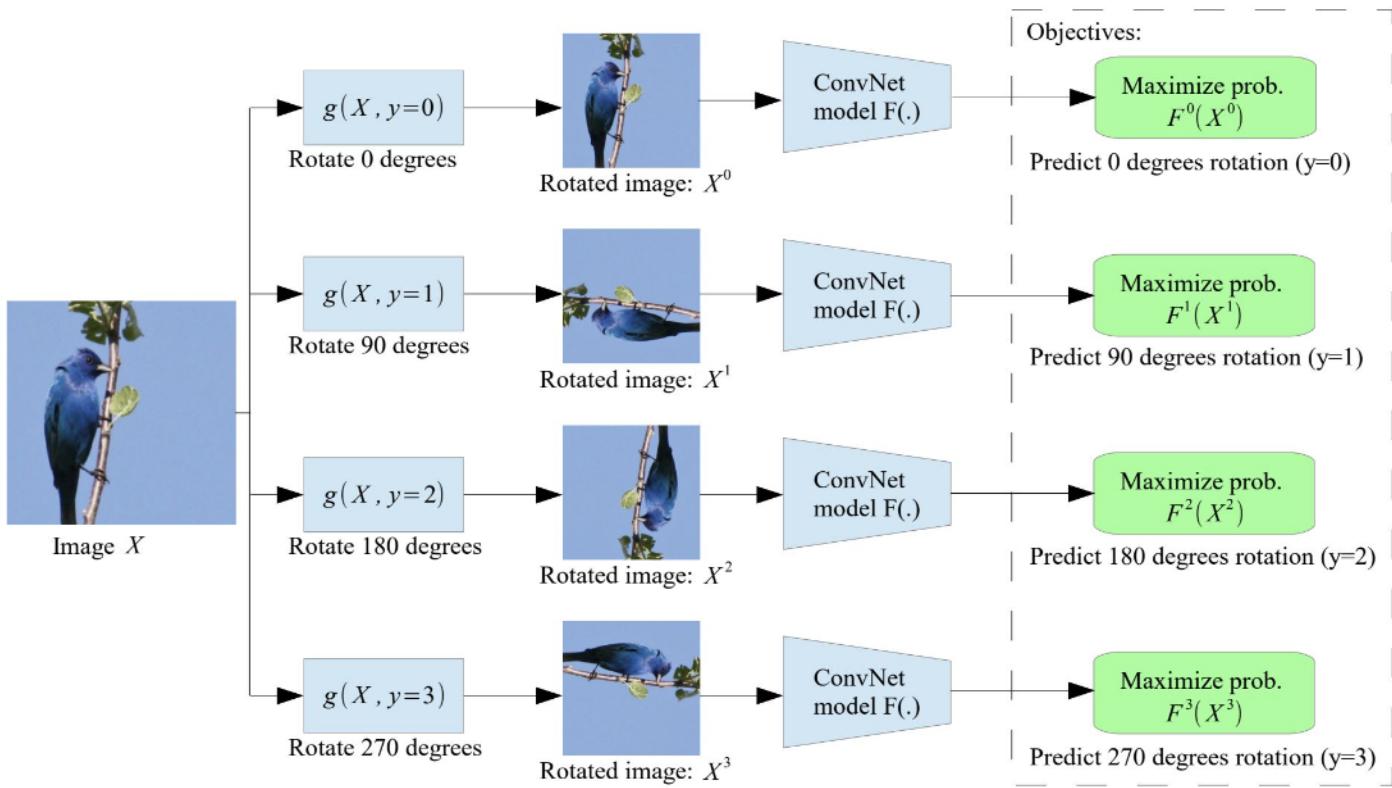


0° rotation



270° rotation

Rotation



Rotation

# Rotations	Rotations	CIFAR-10 Classification Accuracy
4	$0^\circ, 90^\circ, 180^\circ, 270^\circ$	89.06
8	$0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$	88.51
2	$0^\circ, 180^\circ$	87.46
2	$90^\circ, 270^\circ$	85.52

Rotation

Method	Conv4	Conv5
ImageNet labels from (Bojanowski & Joulin, 2017)	59.7	59.7
Random from (Noroozi & Favaro, 2016)	27.1	12.0
Tracking Wang & Gupta (2015)	38.8	29.8
Context (Doersch et al., 2015)	45.6	30.4
Colorization (Zhang et al., 2016a)	40.7	35.2
Jigsaw Puzzles (Noroozi & Favaro, 2016)	45.3	34.6
BIGAN (Donahue et al., 2016)	41.9	32.2
NAT (Bojanowski & Joulin, 2017)	-	36.0
(Ours) RotNet	50.0	43.8

Temporal coherence of color

Task: given a color video ...

Colorize all frames of a gray scale version using a reference frame



Reference Frame



Gray-scale Video

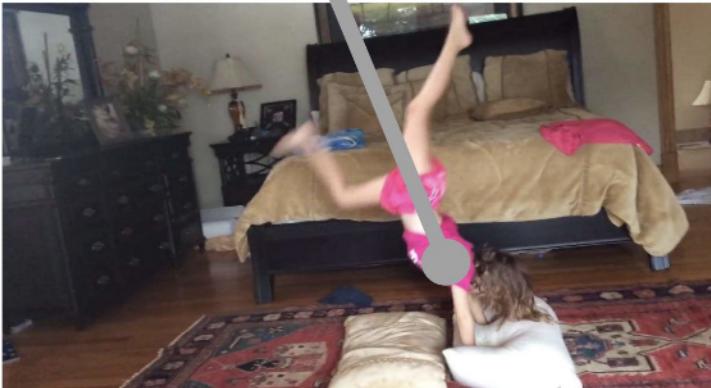
Slide: Zisserman

Temporal coherence of color



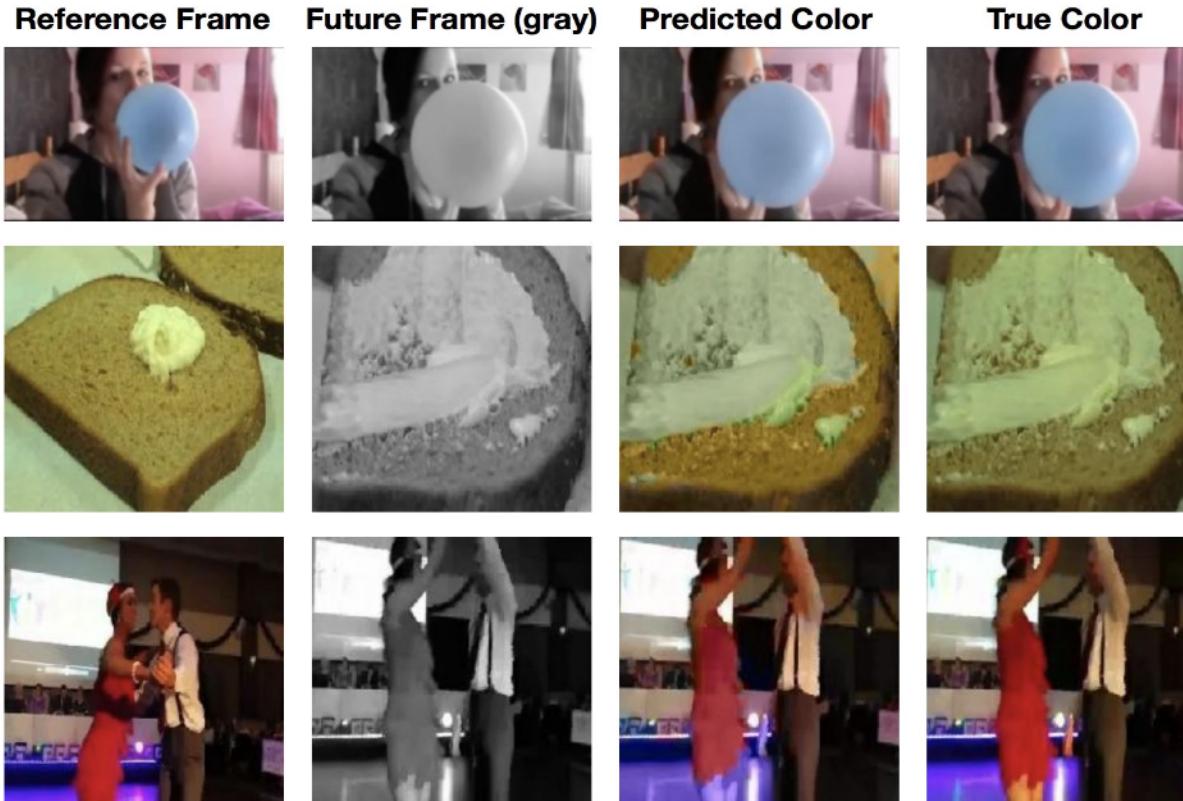
Slide: Zisserman

Temporal coherence of color

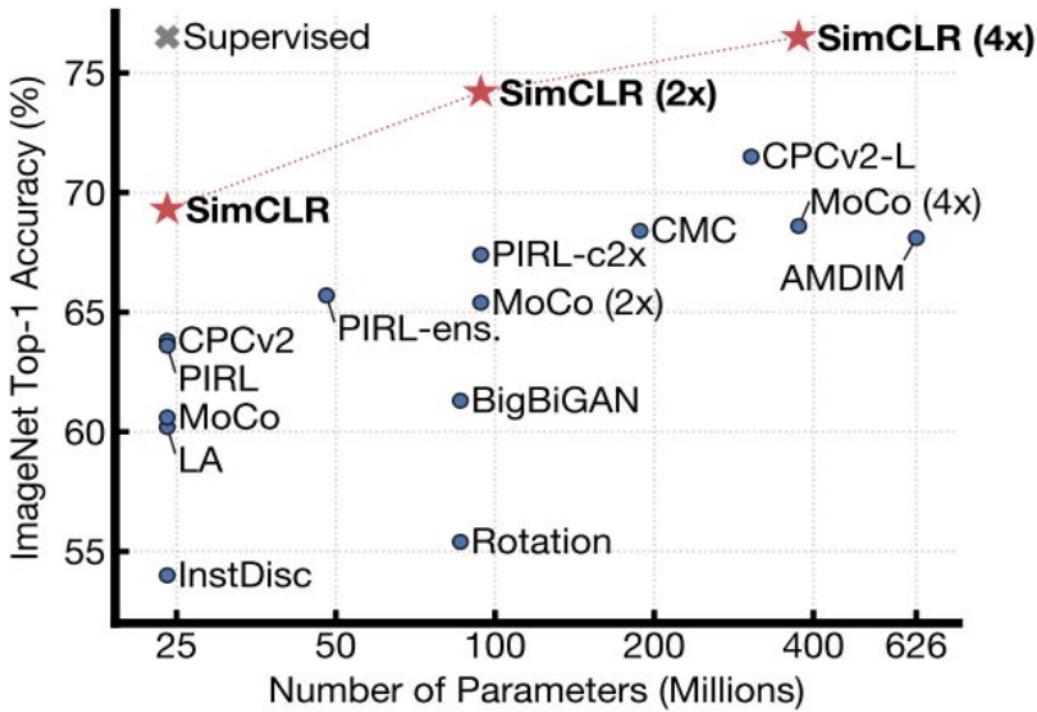


Slide: Zisserman

Tracking emerges from colorization



SimCLR



Instance Discrimination

