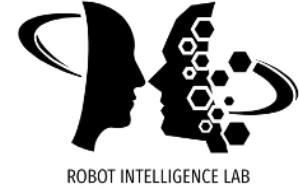


Self-Supervised Learning

Good-old-fashioned SSL

Sungjoon Choi, Korea University

Introduction

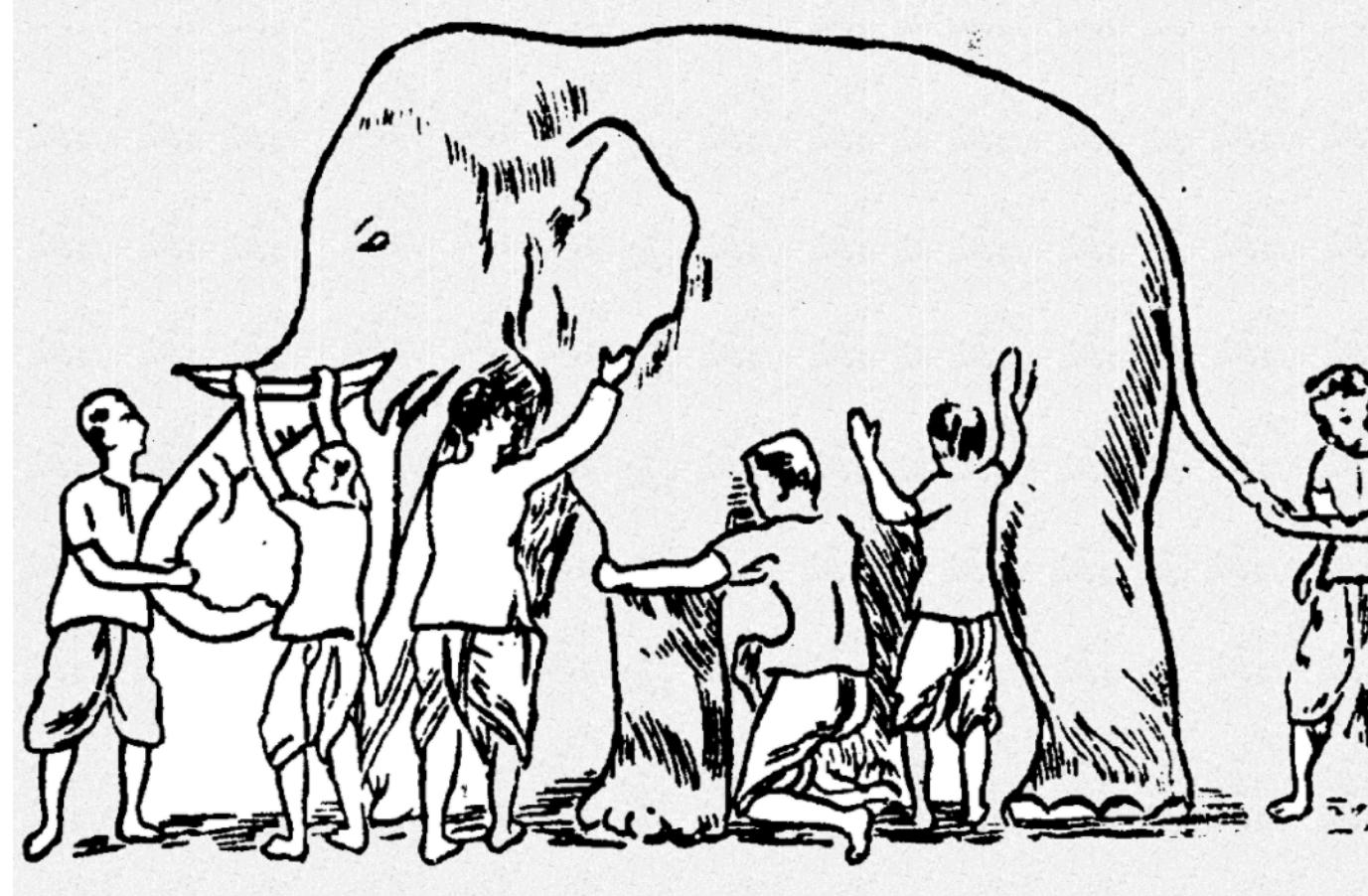


What is self-supervised learning?

Self-Supervised Learning



Caution



Contents

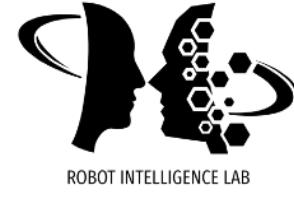
- Jigsaw (2017)
- BiGAN (2017)
- RotNet (2018)
- Auto-Encoding Transformations (2019)
- DeepCluster (2019)
- Single Image SSL (2020)



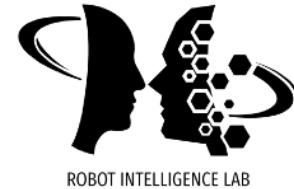
Jigsaw

"Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles," 2017

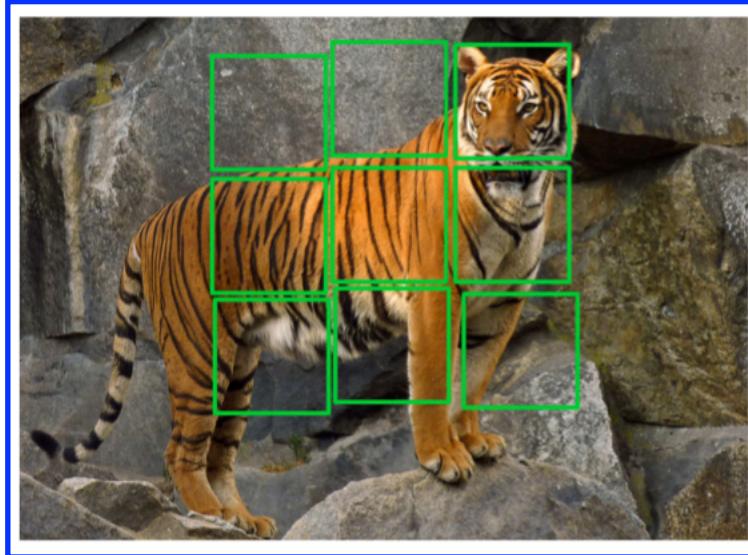
Jigsaw Puzzles



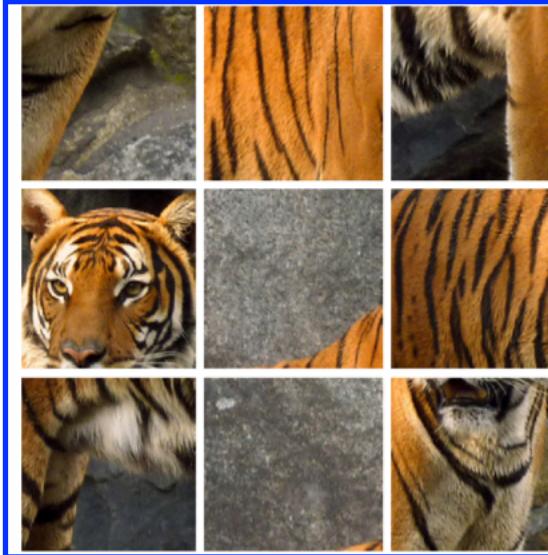
Jigsaw Puzzle Reassembly



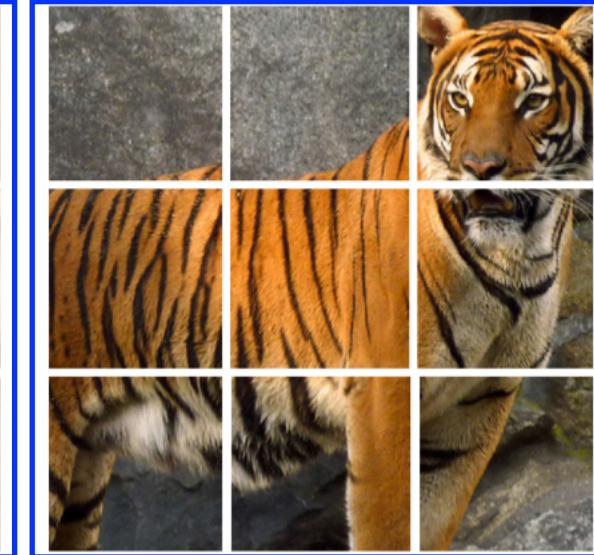
Select patches from an image



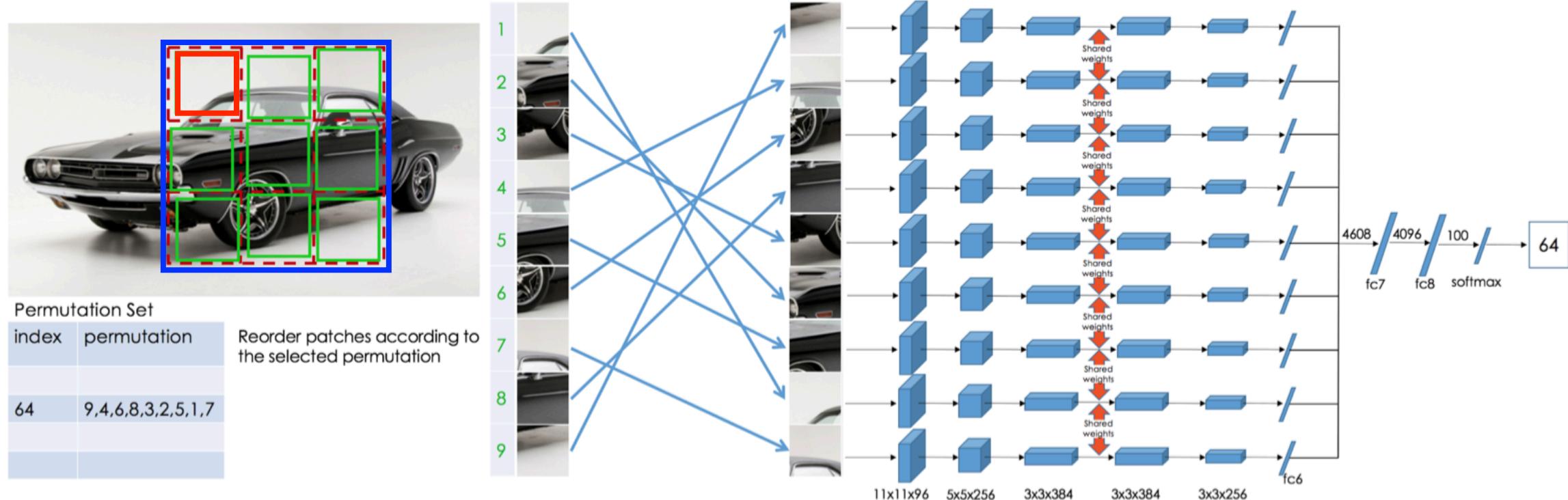
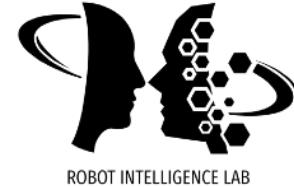
Shuffle patches



Reassemble patches

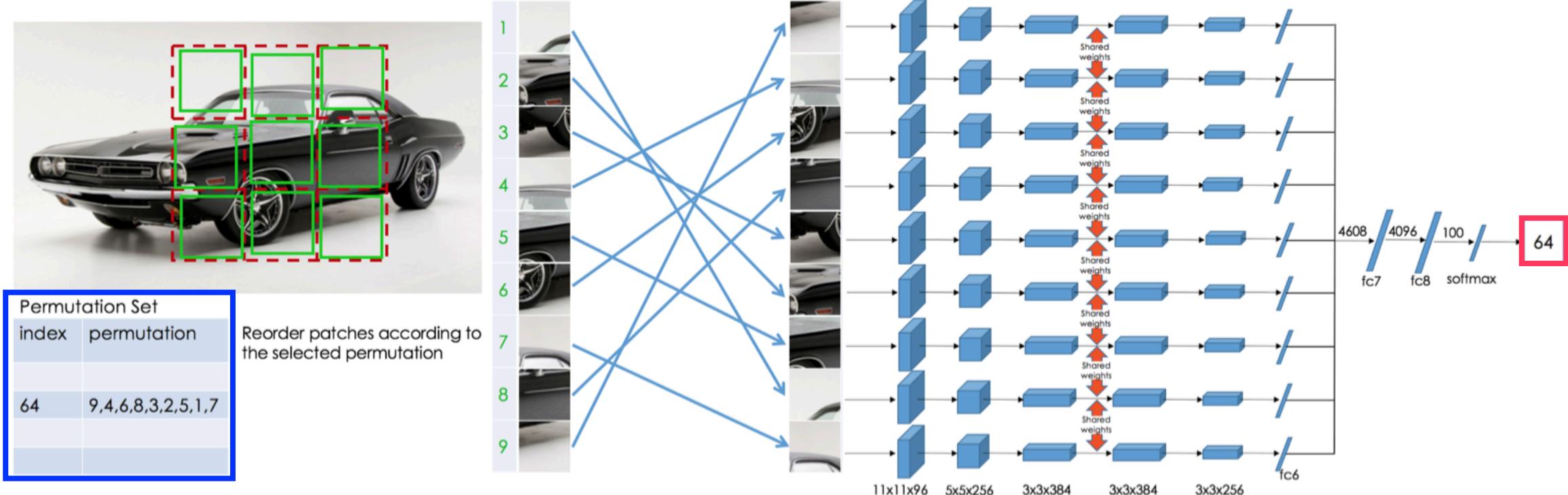
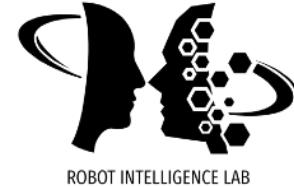


Context-Free Architecture



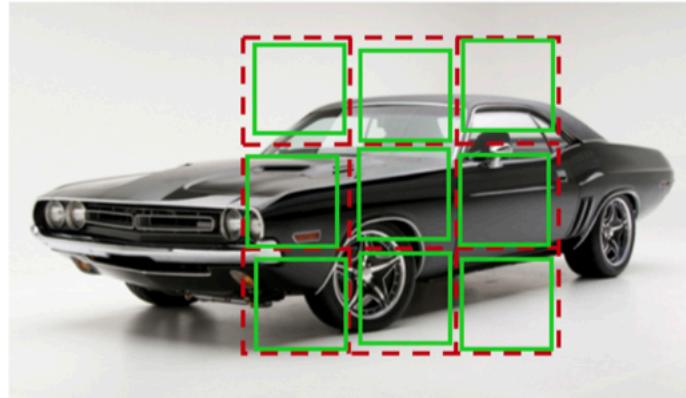
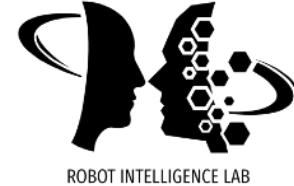
Given an image, context-free architecture randomly crop 225×225 pixel window, and randomly pick a 64×64 pixel tiles from each 75×75 pixel cell.

Context-Free Architecture



The number of total permutation is $9! = 362,880$. However, a subset of a **possible permutation set** is defined (of size 64) and cast the reassembly problem as **a classification problem**.

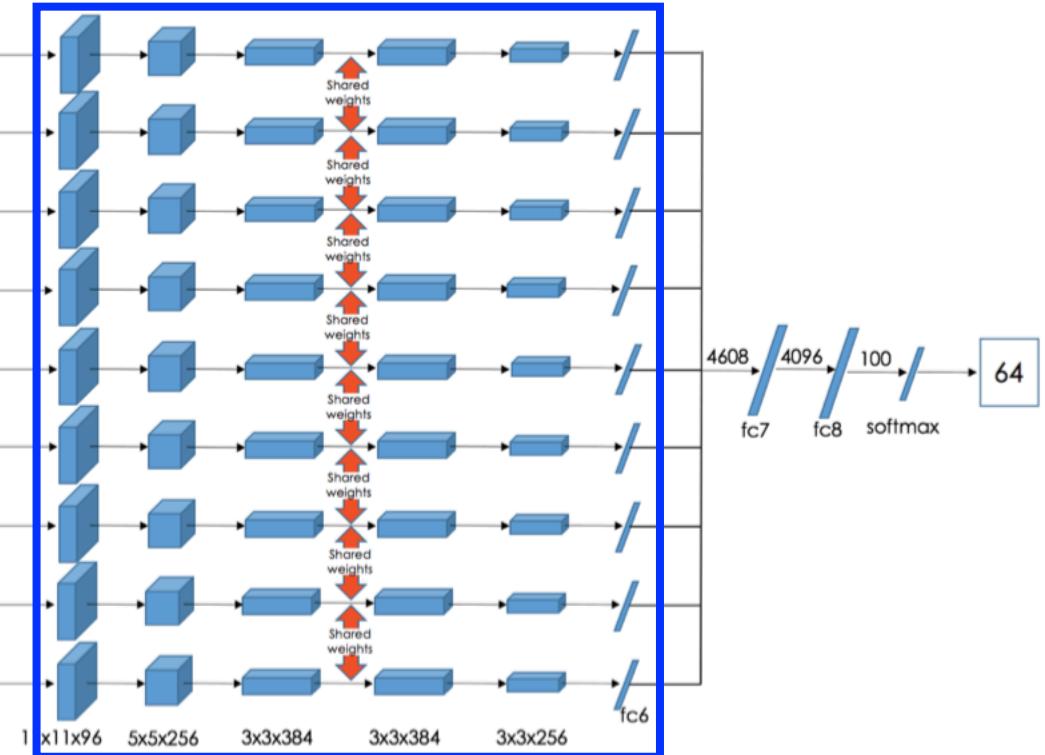
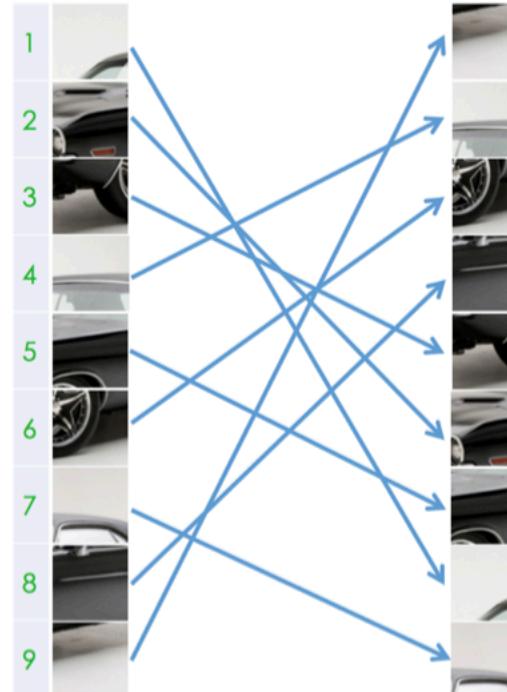
Context-Free Architecture



Permutation Set

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

Reorder patches according to the selected permutation



FCN utilizes AlexNet architecture with **weight-sharing** up-to the *fc6* layer. Nine feature representation vectors are concatenated to predict the permutation class.

Image Retrieval

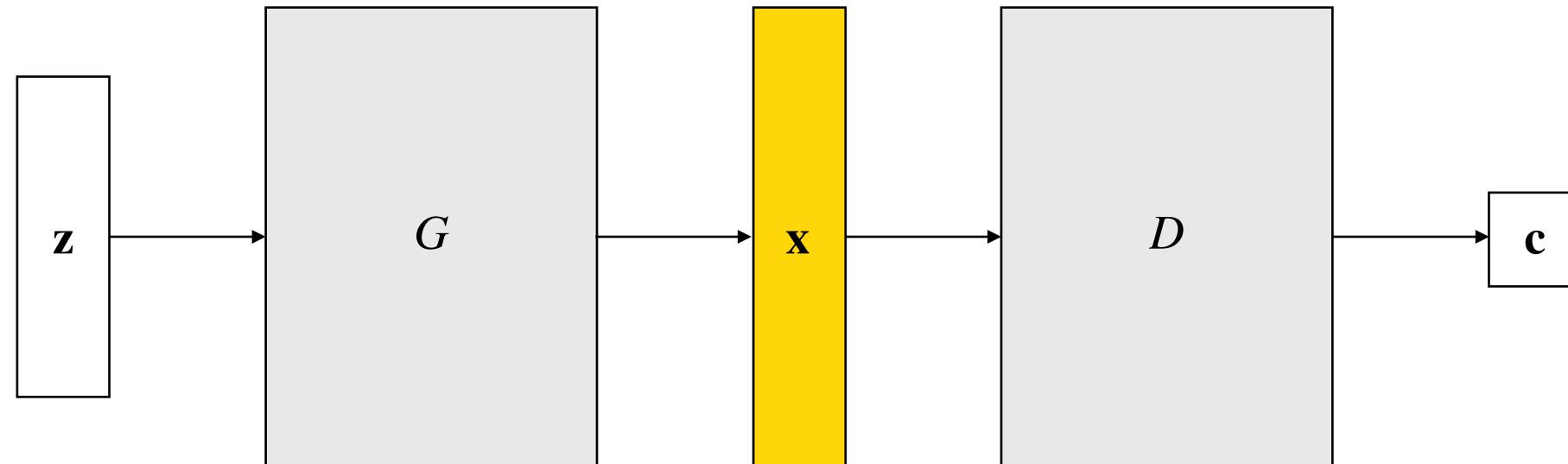




BiGAN

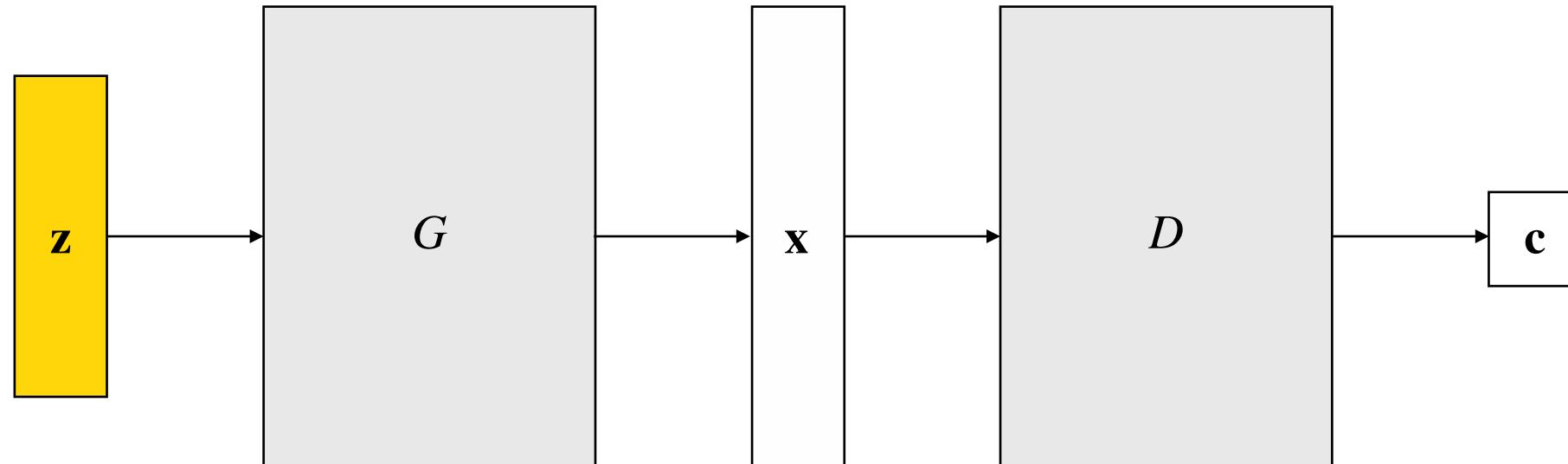
"ADVERSARIAL FEATURE LEARNING," 2017

Generative Adversarial Net



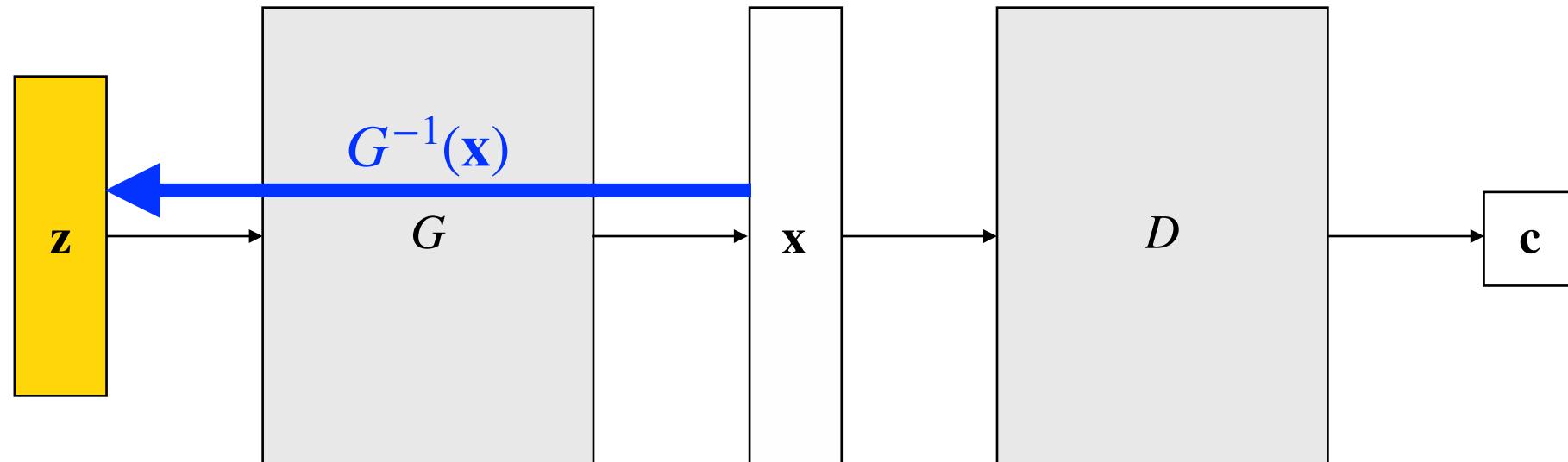
The objective of GAN is to minimize the distance between $p(\mathbf{x})$ and $G(\mathbf{z})$ where \mathbf{z} follows a simple distribution.

Generative Adversarial Net



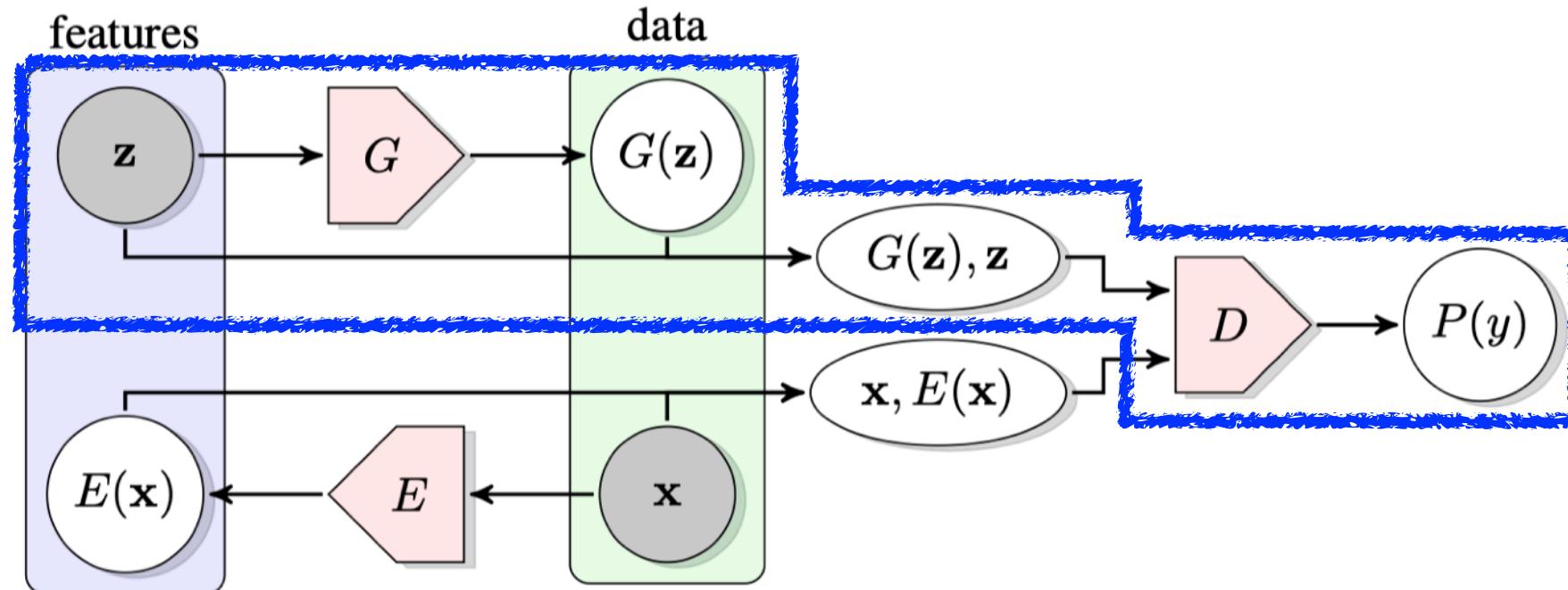
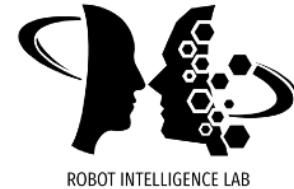
A latent vector \mathbf{z} corresponds to an input \mathbf{x} , which can be thought as meaningful representation of \mathbf{x} .

Generative Adversarial Net



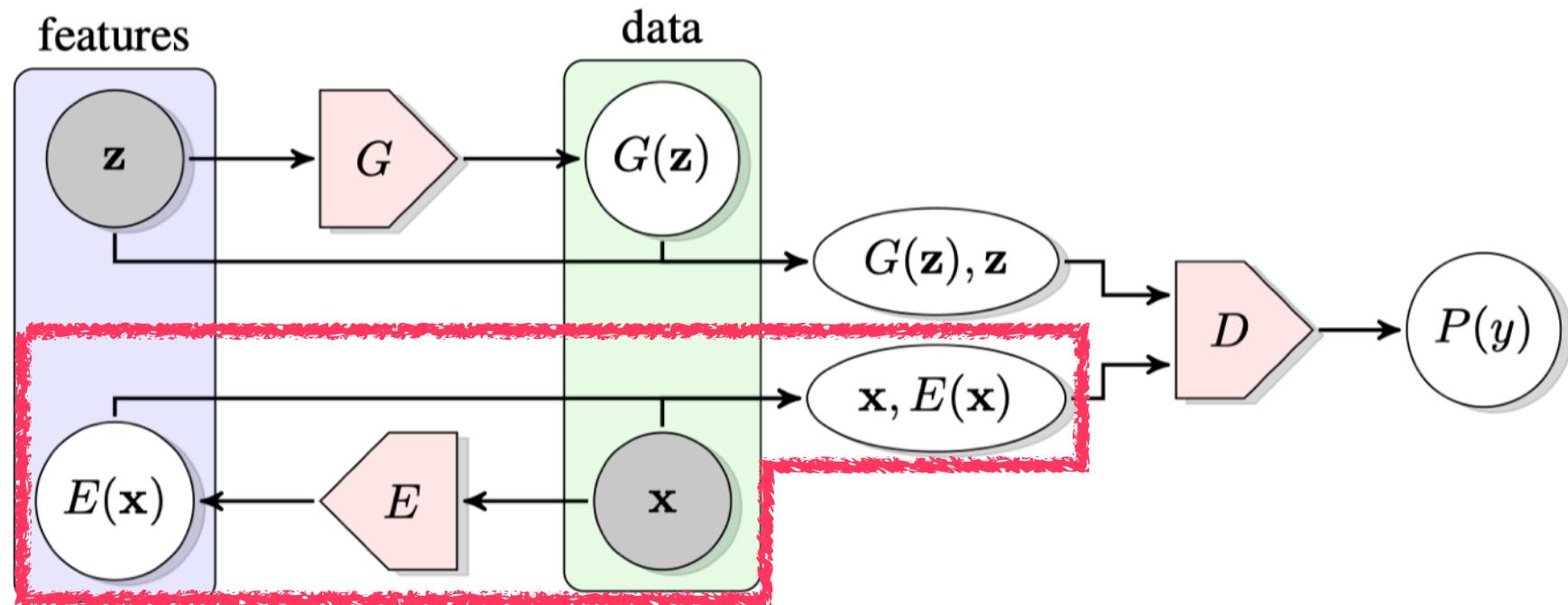
However, there is one remaining problem. G^{-1} does not exist in general.

Bidirectional GAN (BiGAN)



This part is the same as the original GAN

BiGAN



This part is newly added in BiGAN.

BiGAN



The BiGAN training objective is defined as a minimax objective

$$\min_{G,E} \max_D V(D, E, G) \quad (2)$$

where

$$V(D, E, G) := \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \left[\underbrace{\mathbb{E}_{\mathbf{z} \sim p_E(\cdot|\mathbf{x})} [\log D(\mathbf{x}, \mathbf{z})]}_{\log D(\mathbf{x}, E(\mathbf{x}))} \right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \left[\underbrace{\mathbb{E}_{\mathbf{x} \sim p_G(\cdot|\mathbf{z})} [\log (1 - D(\mathbf{x}, \mathbf{z}))]}_{\log(1-D(G(\mathbf{z}), \mathbf{z}))} \right]. \quad (3)$$

The inputs of the **discriminator** are both \mathbf{x} and \mathbf{z} , $D(\mathbf{x}, \mathbf{z})$, (the original discriminator only takes \mathbf{x} , i.e., $D(\mathbf{x})$).

BiGAN



The BiGAN training objective is defined as a minimax objective

$$\min_{G,E} \max_D V(D, E, G) \quad (2)$$

where

$$V(D, E, G) := \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}} \left[\underbrace{\mathbb{E}_{\mathbf{z} \sim p_E(\cdot|\mathbf{x})} [\log D(\mathbf{x}, \mathbf{z})]}_{\log D(\mathbf{x}, E(\mathbf{x}))} \right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} \left[\underbrace{\mathbb{E}_{\mathbf{x} \sim p_G(\cdot|\mathbf{z})} [\log (1 - D(\mathbf{x}, \mathbf{z}))]}_{\log(1 - D(G(\mathbf{z}), \mathbf{z}))} \right]. \quad (3)$$

The newly added **encoder** maps an input \mathbf{x} to a latent vector \mathbf{z} .

BiGAN



The BiGAN training objective is defined as a minimax objective

$$\min_{G,E} \max_D V(D, E, G) \quad (2)$$

where

$$V(D, E, G) := \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \left[\underbrace{\mathbb{E}_{\mathbf{z} \sim p_E(\cdot|\mathbf{x})} [\log D(\mathbf{x}, \mathbf{z})]}_{\log D(\mathbf{x}, E(\mathbf{x}))} \right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \left[\underbrace{\mathbb{E}_{\mathbf{x} \sim p_G(\cdot|\mathbf{z})} [\log (1 - D(\mathbf{x}, \mathbf{z}))]}_{\log(1-D(G(\mathbf{z}), \mathbf{z}))} \right]. \quad (3)$$

To maximize $V(\cdot)$ w.r.t. $D(\cdot)$, the pair of a real image \mathbf{x} and the encoded latent vector $\mathbf{z}' = E(\mathbf{x})$ gets higher value while the pair of a fake image $G(\mathbf{z})$ and the underlying latent vector \mathbf{z} get lower values.

BiGAN



The BiGAN training objective is defined as a minimax objective

$$\min_{G,E} \max_D V(D, E, G) \quad (2)$$

where

$$V(D, E, G) := \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}} \left[\underbrace{\mathbb{E}_{\mathbf{z} \sim p_E(\cdot|\mathbf{x})} [\log D(\mathbf{x}, \mathbf{z})]}_{\log D(\mathbf{x}, E(\mathbf{x})) \downarrow} \right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{z}}} \left[\underbrace{\mathbb{E}_{\mathbf{x} \sim p_G(\cdot|\mathbf{z})} [\log (1 - D(\mathbf{x}, \mathbf{z}))]}_{\log(1-D(G(\mathbf{z}), \mathbf{z}))} \right]. \quad (3)$$

To maximize $V(\cdot)$ w.r.t. $D(\cdot)$, the pair of a real image \mathbf{x} and the encoded latent vector $\mathbf{z}' = E(\mathbf{x})$ gets higher value while the pair of a fake image $G(\mathbf{z})$ and the underlying latent vector \mathbf{z} get lower values.

To minimize $V(\cdot)$ w.r.t. $E(\cdot)$, the encoder $E(\mathbf{x})$ mimics the inverse of $G(\mathbf{z})$, i.e., $G^{-1}(\mathbf{x})$.

BiGAN



The BiGAN training objective is defined as a minimax objective

$$\min_{G,E} \max_D V(D, E, G) \quad (2)$$

where

$$V(D, E, G) := \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{X}}} \left[\underbrace{\mathbb{E}_{\mathbf{z} \sim p_E(\cdot|\mathbf{x})} [\log D(\mathbf{x}, \mathbf{z})]}_{\log D(\mathbf{x}, E(\mathbf{x}))} \right] + \mathbb{E}_{\mathbf{z} \sim p_{\mathbf{Z}}} \left[\underbrace{\mathbb{E}_{\mathbf{x} \sim p_G(\cdot|\mathbf{z})} [\log (1 - D(\mathbf{x}, \mathbf{z}))]}_{\log(1 - D(G(\mathbf{z}), \mathbf{z}))} \right]. \quad (3)$$

An interesting idea of BiGAN is that the training objective of the encoder $E(\mathbf{x})$ is **NOT** directly minimizing $\|E(G(\mathbf{z})) - \mathbf{z}\|_2^2$.

Reconstruction



$$\begin{array}{l} G(\mathbf{z}) \quad \boxed{7 \ 3 \ 6 \ 1 \ 4 \ 2 \ 1 \ 6 \ 1 \ 8 \ 6 \ 6 \ 3 \ 0 \ 2 \ 1 \ 3 \ 4 \ 6 \ 7} \\ \hline \mathbf{x} \quad \boxed{0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9 \ 0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9} \\ G(E(\mathbf{x})) \quad \boxed{0 \ 1 \ 2 \ 3 \ 7 \ 5 \ 1 \ 7 \ 3 \ 7 \ 0 \ 1 \ 2 \ 3 \ 4 \ 4 \ 4 \ 7 \ 8 \ 7} \end{array}$$

Reconstruction



$G(\mathbf{z})$



\mathbf{x}



$G(E(\mathbf{x}))$



\mathbf{x}



$G(E(\mathbf{x}))$



\mathbf{x}



$G(E(\mathbf{x}))$

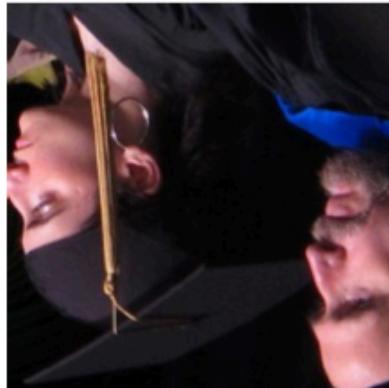
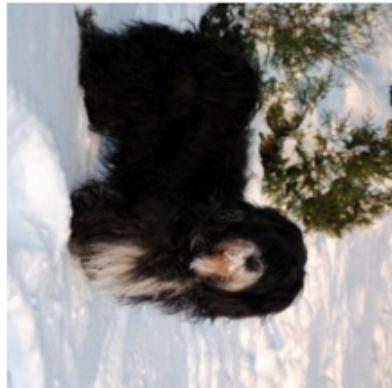
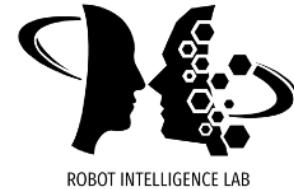




RotNet

"UNSUPERVISED REPRESENTATION LEARNING BY PREDICTING IMAGE ROTATIONS," 2018

Intuition

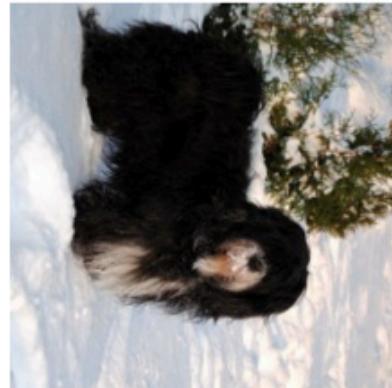


What kind of **supervision** can get out of these images?

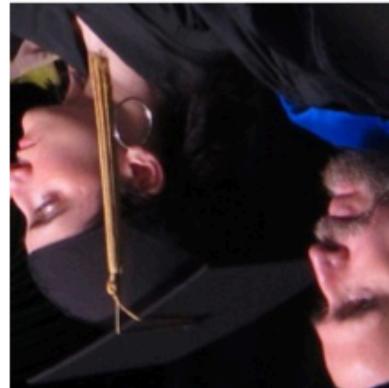
Intuition



90° rotation



270° rotation



180° rotation



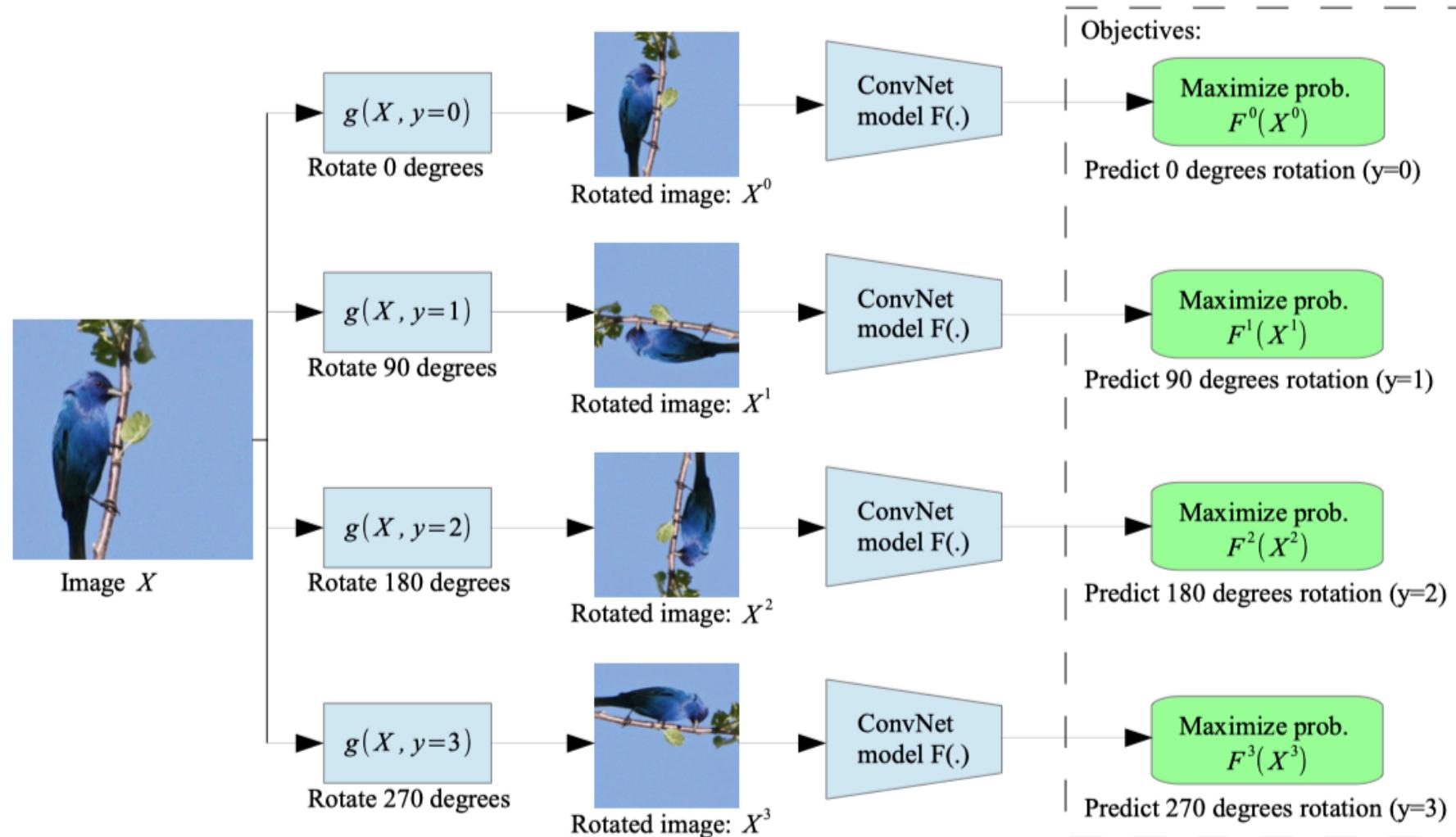
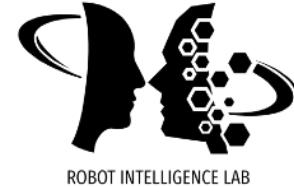
0° rotation



270° rotation

Predict the **rotated angle** of the original image!

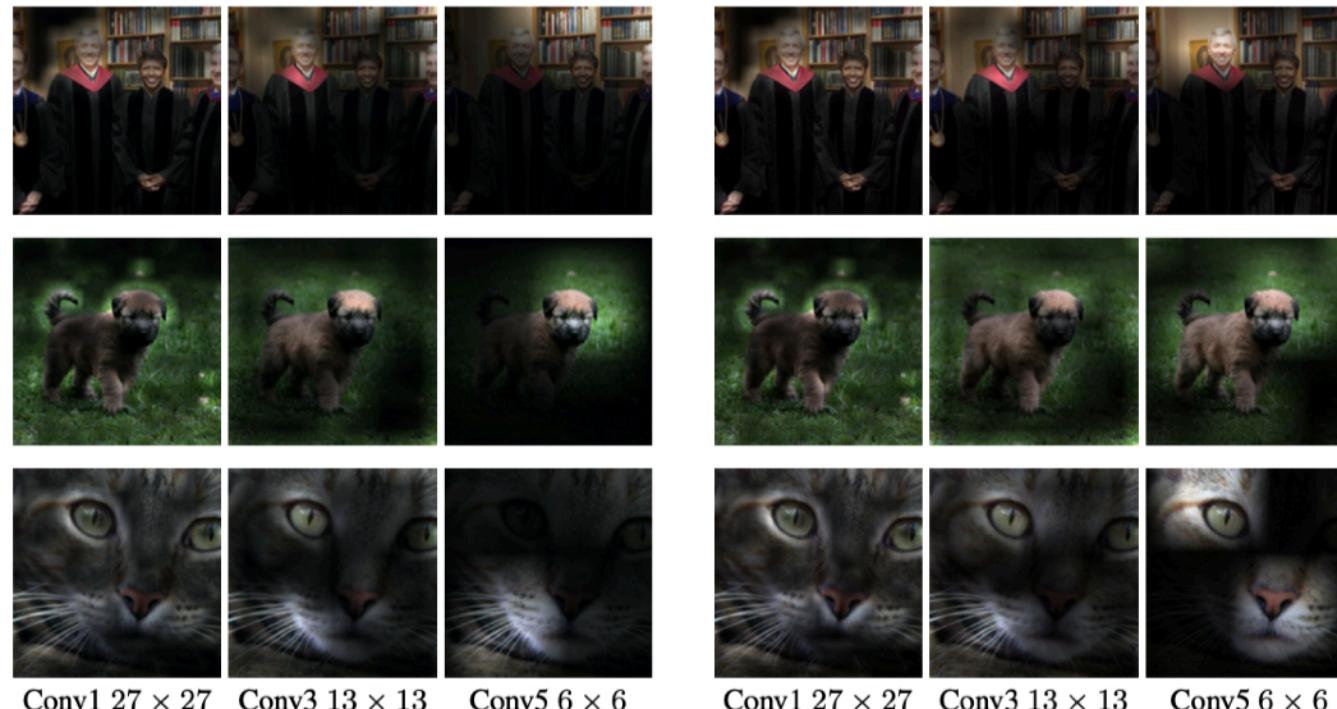
Self-supervised Task



Attention Maps



Input images on the models



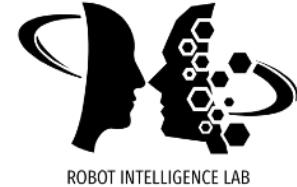
(a) Attention maps of supervised model

(b) Attention maps of our self-supervised model

*Attention map is computed based on the magnitude of activations at each spatial cell.

Experiments

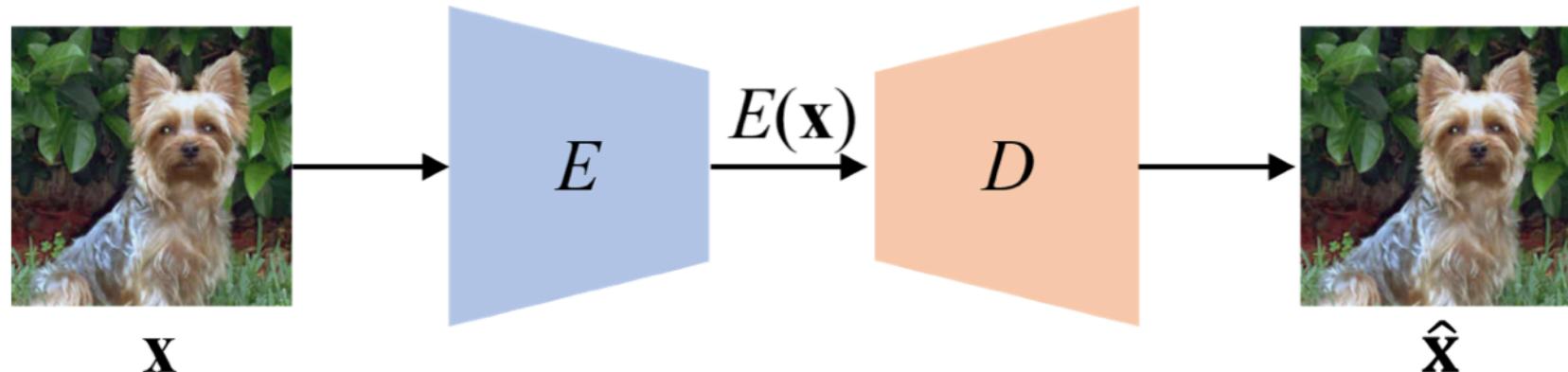
- Multiple experiments including unsupervised feature learning for
 - ImageNet classification
 - PASCAL classification
 - PASCAL detection
 - PASCAL segmentation
 - CIFAR-10 classification



Auto-Encoding Transformations

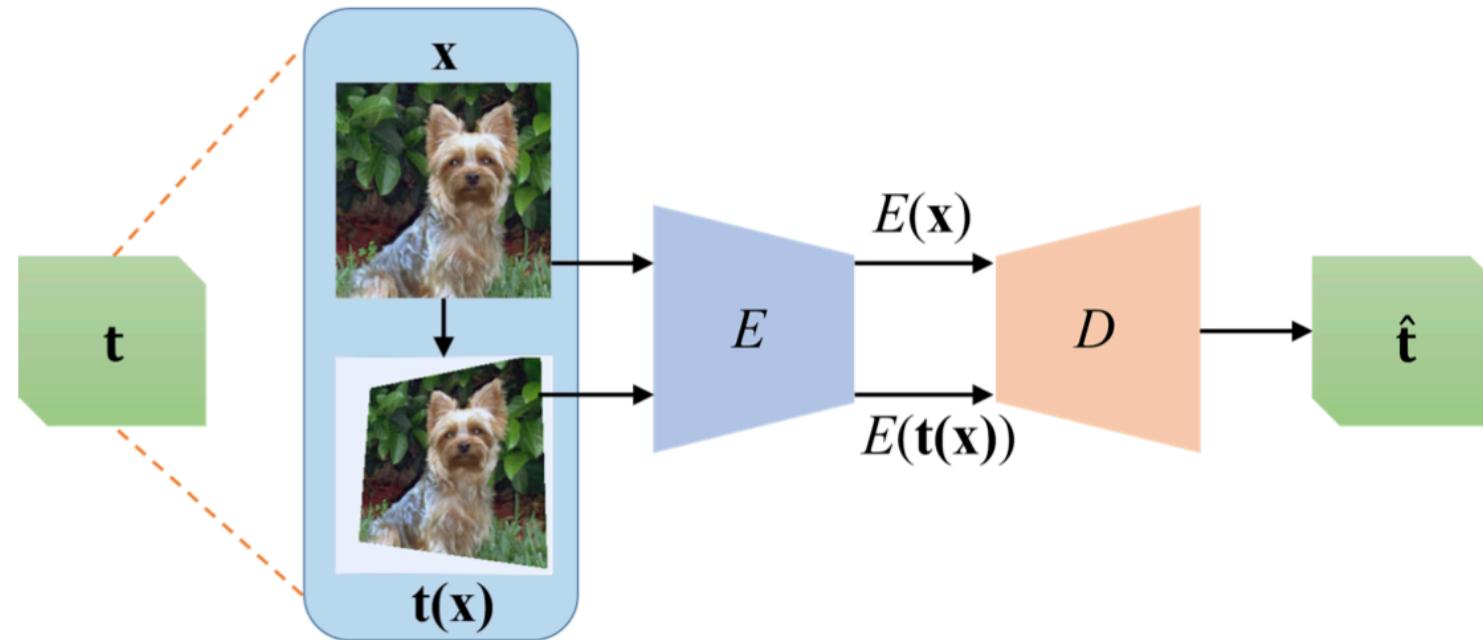
"AET vs. AED: Unsupervised Representation Learning by Auto-Encoding Transformations rather than Data," 2019

Auto-Encoders



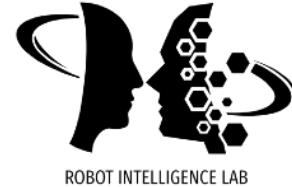
Auto-Encoding Data (AED) has been widely used for unsupervised learning.

Auto-Encoding Transformation



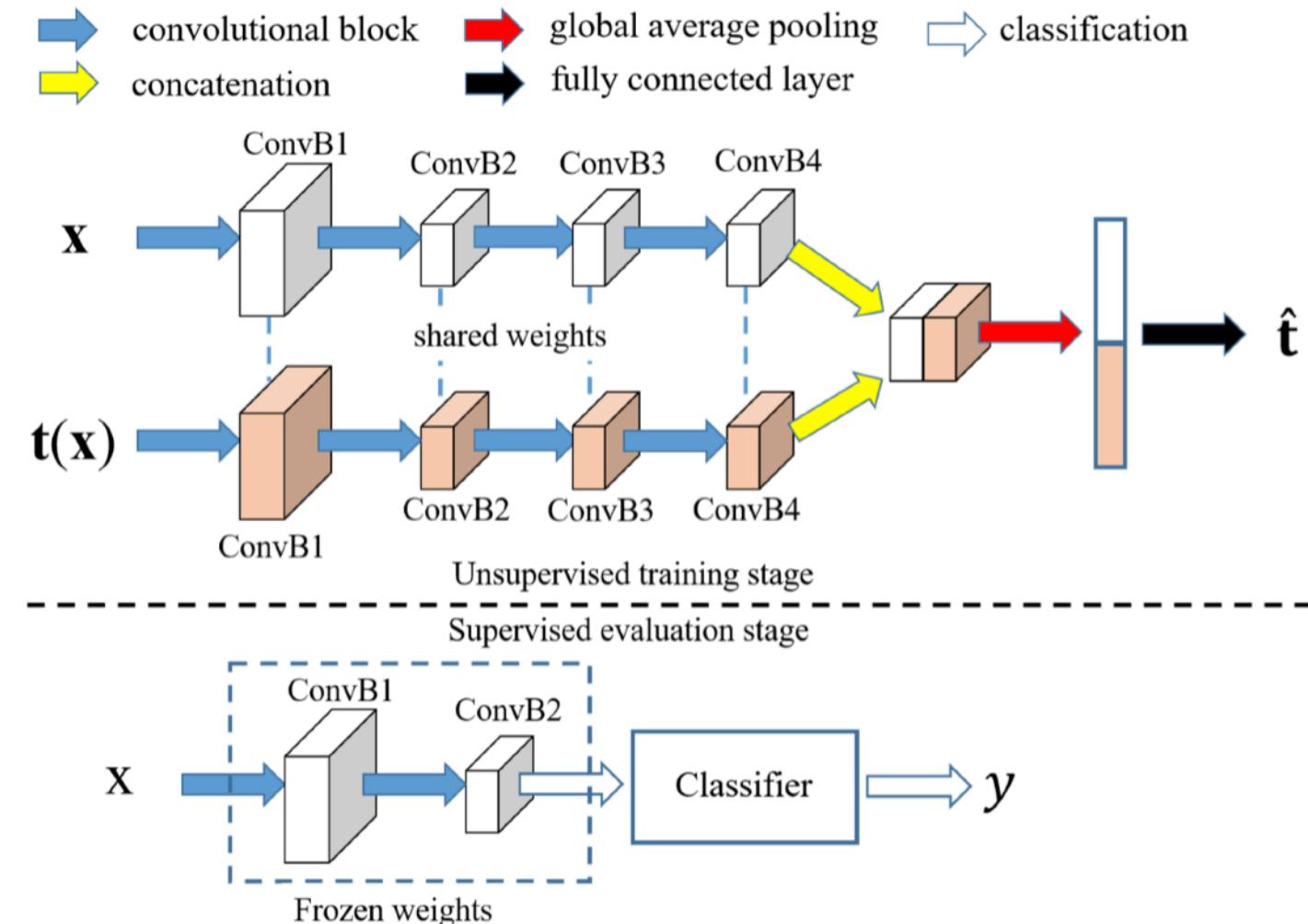
This paper presents **auto-encoding transformation (AET)** as a mean of self-supervised learning.

Problem Formulation

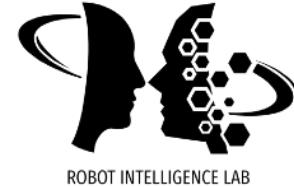


- Encoder
 - $E : \mathbf{x} \mapsto E(\mathbf{x})$ extracts the **representation** of a sample (image)
- Transformation
 - $\mathbf{t} : \mathbf{x} \mapsto \mathbf{t}(\mathbf{x})$ where a transformation is sampled from a distribution \mathcal{T}
 - The transform distribution $\mathcal{T} = \{\mathbf{t}_\theta | \theta \sim \Theta\}$ is parametrized by θ .
 - Projective and affine transformations: $M(\theta) \in \mathbb{R}^{3 \times 3}$ where the loss becomes
$$l(\mathbf{t}_\theta, \hat{\mathbf{t}}_\theta) = \frac{1}{2} \|M(\theta) - M(\hat{\theta})\|_2^2$$
 - GAN-Induced transformations: $\mathbf{t}_z(\mathbf{x}) = G(\mathbf{x}, \mathbf{z})$ where the loss becomes
$$l(\mathbf{t}_z - \hat{\mathbf{t}}_z) = \frac{1}{2} \|\mathbf{z} - \hat{\mathbf{z}}\|_2^2$$
- Decoder
 - $D : [E(\mathbf{x}), E(\mathbf{t}(\mathbf{x}))] \mapsto \hat{\mathbf{t}}$ predicts the **input transformation** from the original and transformed images.

Network Architectures



CIFAR-10



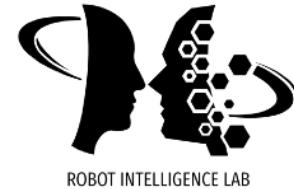
Method	Error rate
Supervised NIN (Lower Bound)	7.20
Random Init. + conv (Upper Bound)	27.50
Roto-Scat + SVM [22]	17.7
ExamplarCNN [7]	15.7
DCGAN [26]	17.2
Scattering [21]	15.3
RotNet + FC [10]	10.94
RotNet + conv [10]	8.84
(Ours) AET-affine + FC	9.77
(Ours) AET-affine + conv	8.05
(Ours) AET-project + FC	9.41
(Ours) AET-project + conv	7.82

ImageNet (Top-1 Accuracy)



Method	Conv4	Conv5
ImageNet Labels [3] (Upper Bound)	59.7	59.7
Random [20] (Lower Bound)	27.1	12.0
Tracking [29]	38.8	29.8
Context [5]	45.6	30.4
Colorization [31]	40.7	35.2
Jigsaw Puzzles [19]	45.3	34.6
BiGAN [6]	41.9	32.2
NAT [3]	-	36.0
DeepCluster [4]	-	44.0
RotNet [10]	50.0	43.8
(Ours) AET-project	53.2	47.0

Reconstruction Results

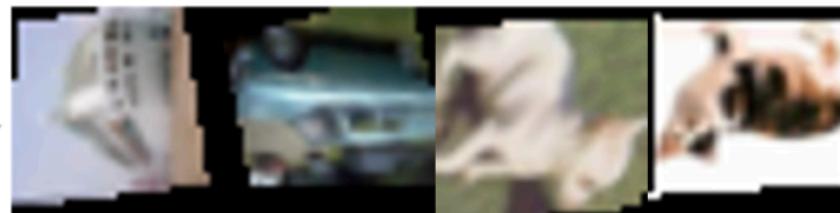


CIFAR-10

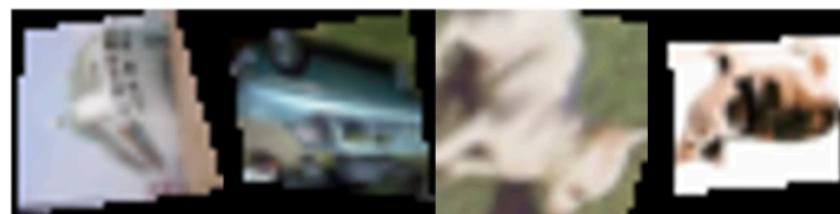
Original



Transformed



Estimated



ImageNet

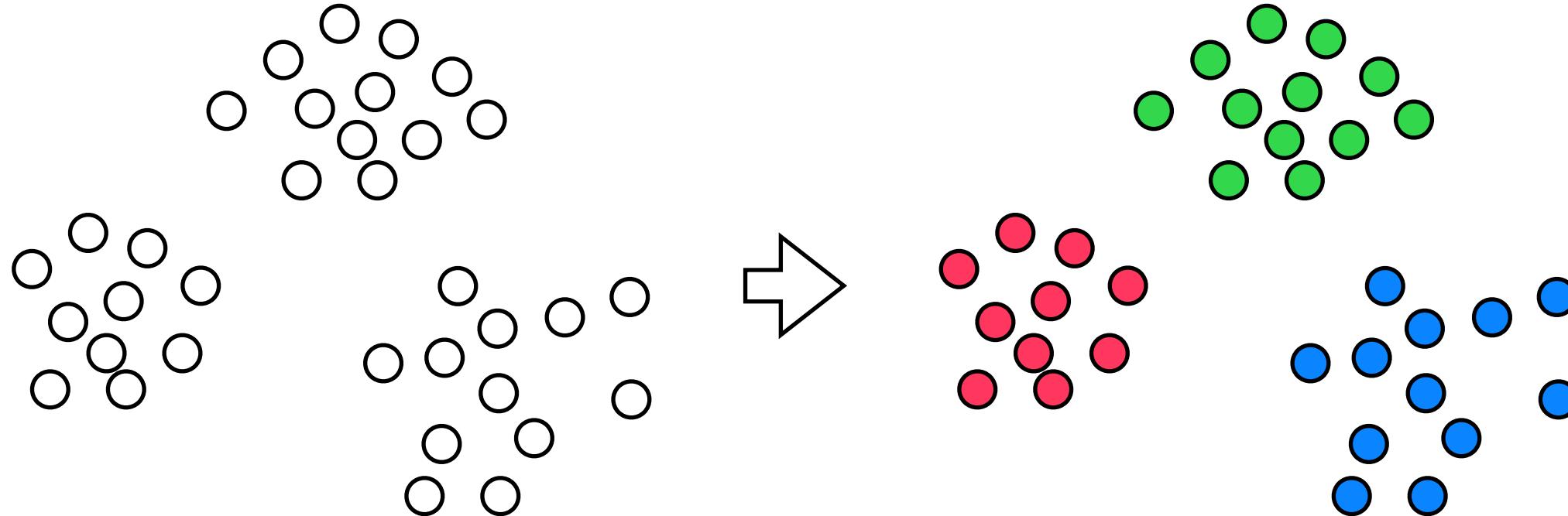
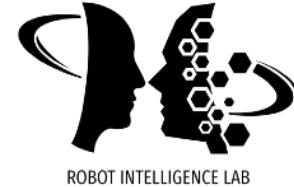




DeepCluster

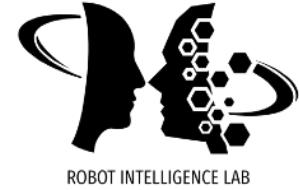
"Deep Clustering for Unsupervised Learning of Visual Features," 2019

Clustering



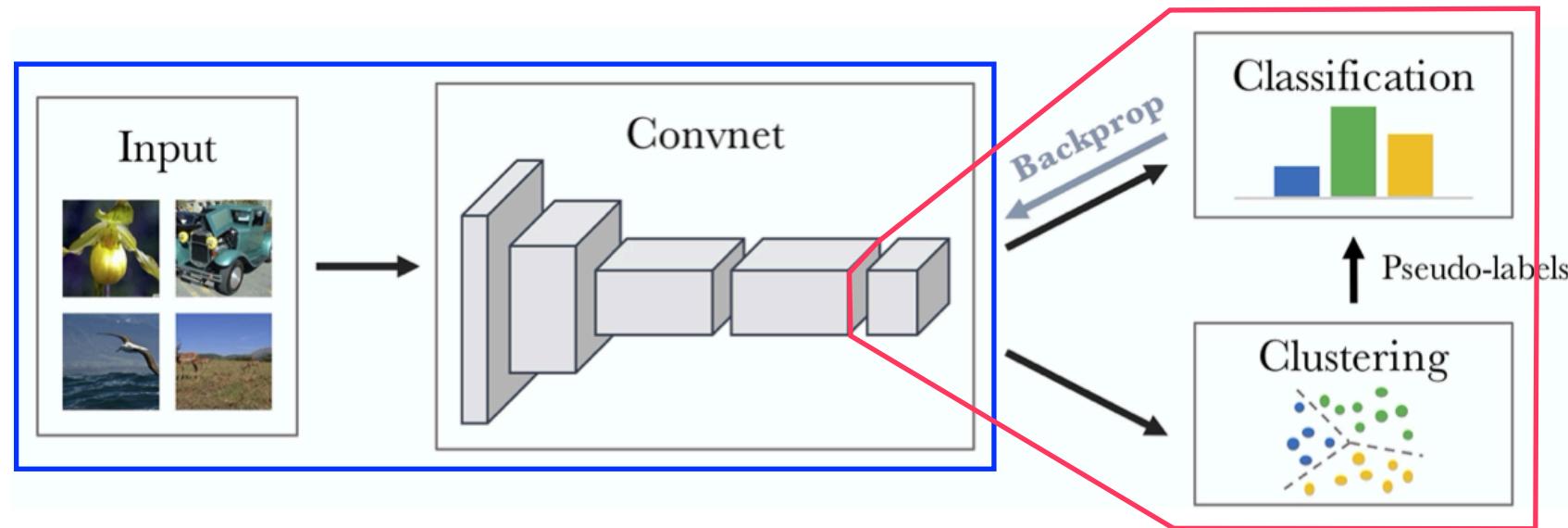
Conventional clustering methods work fairly well on **low-dimensional data** where **distance measures** can easily be defined and evaluated.

Distance Measure



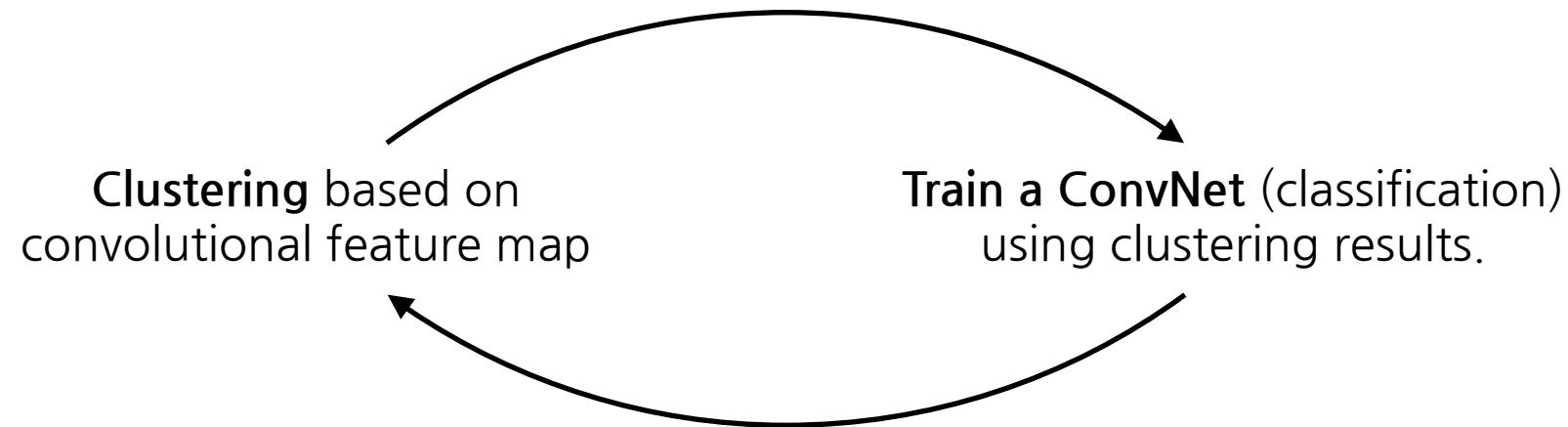
However, in high-dimensional space, computing distances between instances becomes **unintuitive**.

Deep Clustering



This paper combines the **representation learning power** of a Convnet with an **auxiliary learning objectives** from clustering results.

Deep Clustering



Results



BiGAN
Jigsaw

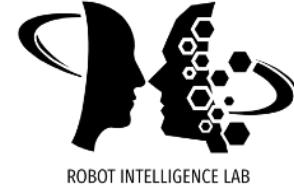
Method	ImageNet					Places				
	conv1	conv2	conv3	conv4	conv5	conv1	conv2	conv3	conv4	conv5
Places labels	—	—	—	—	—	22.1	35.1	40.2	43.3	44.6
ImageNet labels	19.3	36.3	44.2	48.3	50.5	22.7	34.8	38.4	39.4	38.7
Random	11.6	17.1	16.9	16.3	14.1	15.7	20.3	19.8	19.1	17.5
Pathak <i>et al.</i> [38]	14.1	20.7	21.0	19.8	15.5	18.2	23.2	23.4	21.9	18.4
Doersch <i>et al.</i> [25]	16.2	23.3	30.2	31.7	29.6	19.7	26.7	31.9	32.7	30.9
Zhang <i>et al.</i> [28]	12.5	24.5	30.4	31.5	30.3	16.0	25.7	29.6	30.3	29.7
Donahue <i>et al.</i> [20]	17.7	24.5	31.0	29.9	28.0	21.4	26.2	27.1	26.1	24.0
Noroozi and Favaro [26]	18.2	28.8	34.0	33.9	27.1	23.0	32.1	35.5	34.8	31.3
Noroozi <i>et al.</i> [45]	18.0	30.6	34.3	32.5	25.7	23.3	33.9	36.3	34.7	29.6
Zhang <i>et al.</i> [43]	17.7	29.3	35.4	35.2	32.8	21.3	30.7	34.0	34.1	32.5
DeepCluster	12.9	29.2	38.2	39.8	36.1	18.6	30.8	37.0	37.5	33.1



Single Image SSL

"A CRITICAL ANALYSIS OF SELF-SUPERVISION , WHAT WE CAN LEARN FROM A SINGLE IMAGE," 2020

Question



Is self-supervision **able to exploit** the information contained in a large number of images in order to learn different parts of a neural network?

Findings



- 1.** A single image is sufficient, when combined with self-supervision and data augmentation, to learn the **first few layers** of standard deep network as well as using millions of images and full supervision.

- 2.** For the deeper layers of the network, self-supervision remains inferior to supervision even if millions of images are used for training.

SSL Methods



- BiGAN ("ADVERSARIAL FEATURE LEARNING," 2017)
- RotNet ("UNSUPERVISED REPRESENTATION LEARNING BY PREDICTING IMAGE ROTATIONS," 2018)
- DeepCluster ("Deep Clustering for Unsupervised Learning of Visual Features," 2019)

Data



- **Baseline:** trained on d source images (e.g., $d = 1,281,167$ for ImageNet)
- **SSL methods:** $N \ll d$ source images (i.i.d. samples) and the remaining $d - N$ images are augmentations of the source images
 - N controls the **amount of information** in the training data, and if $N = 1$, we are using a single image for training.
- **Augmentations:** cropping, scaling, rotation, contrast changes, and adding noise

Real Samples

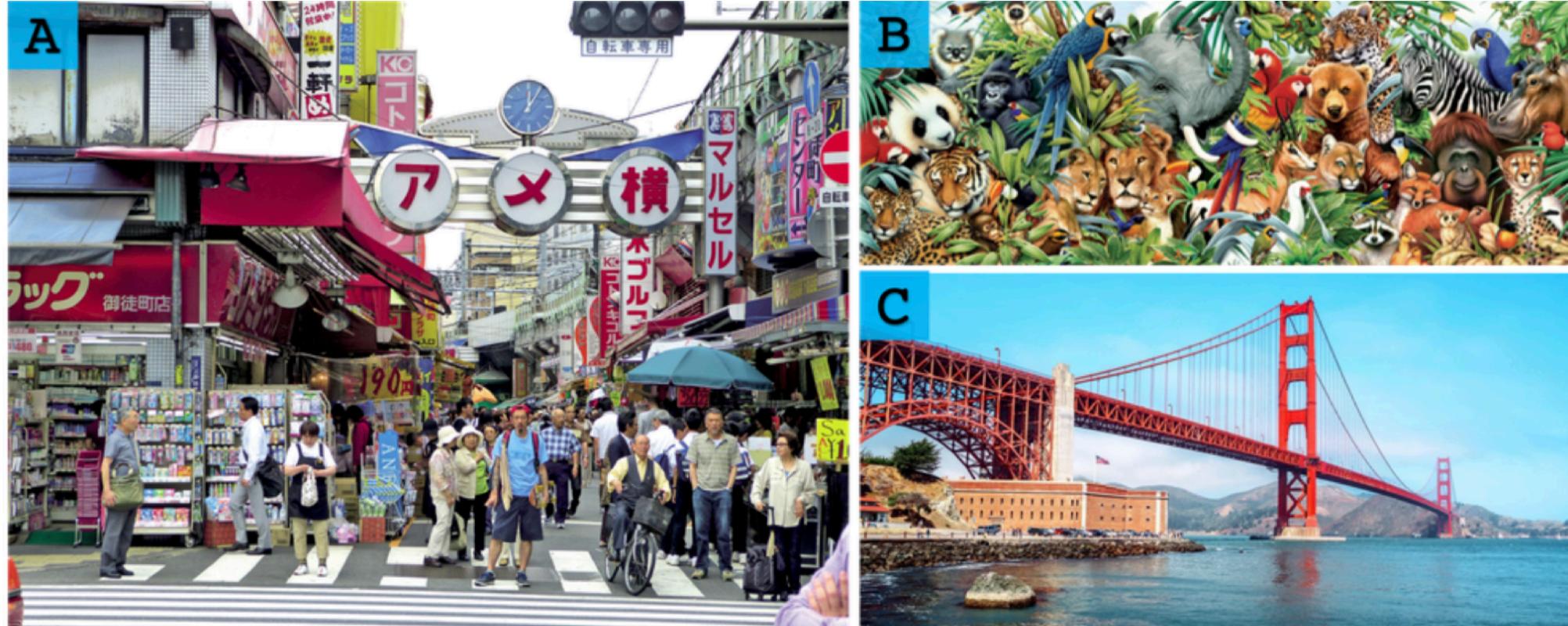
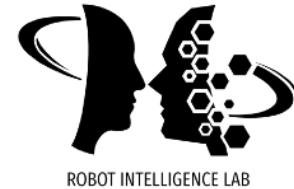
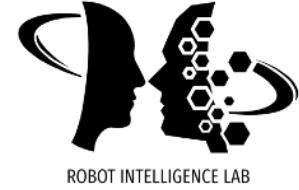


Image A and B: contain rich texture

Image C: ablates "crowdness" as it contains large areas covering no objects

Linear Probes



Solve a difficult task such as ImageNet classification by training a **linear classifier** on top of pre-trained feature representations, which are kept fixed.

ImageNet

Method, Reference	#images	ILSVRC-12				
		conv1	conv2	conv3	conv4	conv5
(a) Full-supervision [‡]	1,281,167	19.3	36.3	44.2	48.3	50.5
(b) (Oyallon et al., 2017): Scattering	0	-	18.9	-	-	-
(c) Random [‡]	0	11.6	17.1	16.9	16.3	14.1
(d) (Krähenbühl et al., 2016): k-means [‡]	≈ 160	17.5	23.0	24.5	23.2	20.6
(e) (Donahue et al., 2017): BiGAN [‡]	1,281,167	17.7	24.5	31.0	29.9	28.0
(f) mono, Image A	1	20.4	30.9	33.4	28.4	16.0
(g) mono, Image B	1	20.5	30.4	31.6	27.0	16.8
(h) deka	10	16.2	16.5	16.5	13.1	7.5
(i) kilo	1,000	16.1	17.7	18.3	17.6	13.5
(j) (Gidaris et al., 2018): RotNet	1,281,167	18.8	31.7	38.7	38.2	36.5
(k) mono, Image A	1	19.9	30.2	30.6	27.6	21.9
(l) mono, Image B	1	17.8	27.6	27.9	25.4	20.2
(m) deka	10	19.6	30.7	32.6	28.9	22.6
(n) kilo	1,000	21.0	33.5	36.5	34.0	29.4
(o) (Caron et al., 2018): DeepCluster	1,281,167	18.0	32.5	39.2	37.2	30.6
(p) mono, Image A	1	20.7	31.5	32.5	28.5	21.0
(q) mono, Image B	1	19.7	30.1	31.6	28.5	20.4
(r) mono, Image C	1	18.9	29.2	31.5	28.9	23.5
(s) deka	10	18.5	29.0	31.1	28.2	21.9
(t) kilo	1,000	19.5	29.8	33.0	31.7	26.8

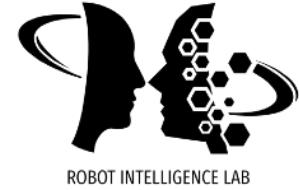
On lower layers, the performance gap between fully-supervised and self-supervised methods is small.

ImageNet

Method, Reference		#images	conv1	conv2	conv3	ILSVRC-12	
						conv4	conv5
(a)	Full-supervision [‡]	1,281,167	19.3	36.3	44.2	48.3	50.5
(b)	(Oyallon et al., 2017): Scattering	0	-	18.9	-	-	-
(c)	Random [‡]	0	11.6	17.1	16.9	16.3	14.1
(d)	(Krähenbühl et al., 2016): <i>k</i> -means [‡]	≈ 160	17.5	23.0	24.5	23.2	20.6
(e)	(Donahue et al., 2017): BiGAN [‡]	1,281,167	17.7	24.5	31.0	29.9	28.0
(f)	mono, Image A	1	20.4	30.9	33.4	28.4	16.0
(g)	mono, Image B	1	20.5	30.4	31.6	27.0	16.8
(h)	deka	10	16.2	16.5	16.5	13.1	7.5
(i)	kilo	1,000	16.1	17.7	18.3	17.6	13.5
(j)	(Gidaris et al., 2018): RotNet	1,281,167	18.8	31.7	38.7	38.2	36.5
(k)	mono, Image A	1	19.9	30.2	30.6	27.6	21.9
(l)	mono, Image B	1	17.8	27.6	27.9	25.4	20.2
(m)	deka	10	19.6	30.7	32.6	28.9	22.6
(n)	kilo	1,000	21.0	33.5	36.5	34.0	29.4
(o)	(Caron et al., 2018): DeepCluster	1,281,167	18.0	32.5	39.2	37.2	30.6
(p)	mono, Image A	1	20.7	31.5	32.5	28.5	21.0
(q)	mono, Image B	1	19.7	30.1	31.6	28.5	20.4
(r)	mono, Image C	1	18.9	29.2	31.5	28.9	23.5
(s)	deka	10	18.5	29.0	31.1	28.2	21.9
(t)	kilo	1,000	19.5	29.8	33.0	31.7	26.8

However, on higher layers, SSL methods show inferior performance.

Conclusion



SSL with a single image can successfully capture the simplest image statistics (lower level features), but a gap exists with full supervision even when using large datasets.

Conclusion



Good old fashioned SSL methods, more to come next week!

Thank You



ROBOT INTELLIGENCE LAB