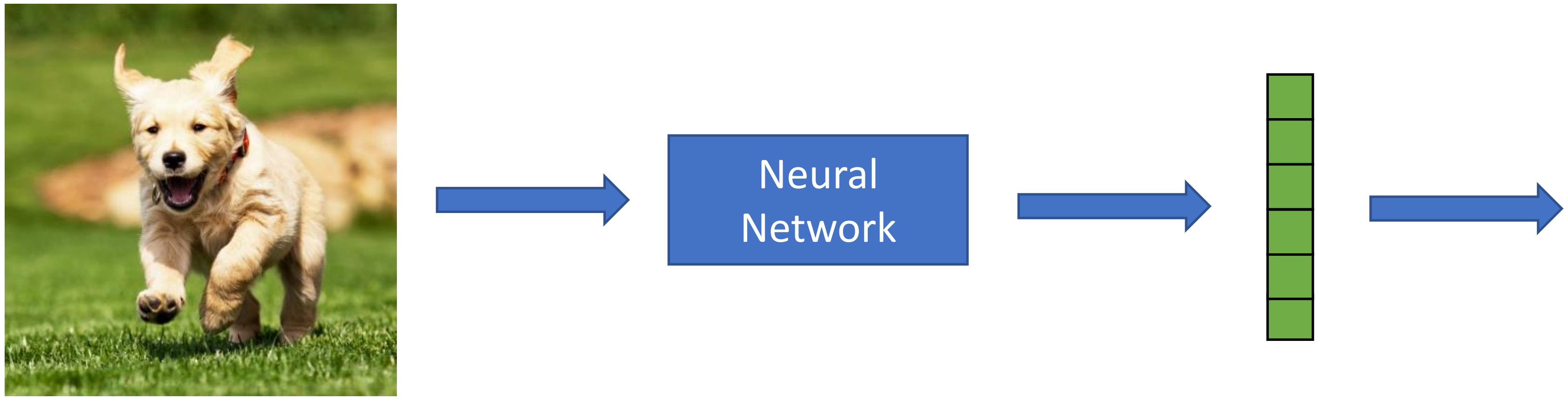
Visual Representation Learning

CS 6384 Computer Vision

Professor Yu Xiang

The University of Texas at Dallas

Learning Visual Representations

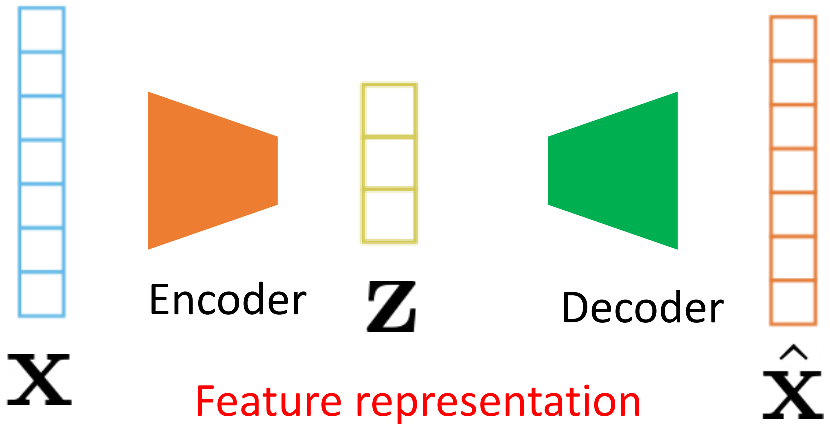


Feature representation

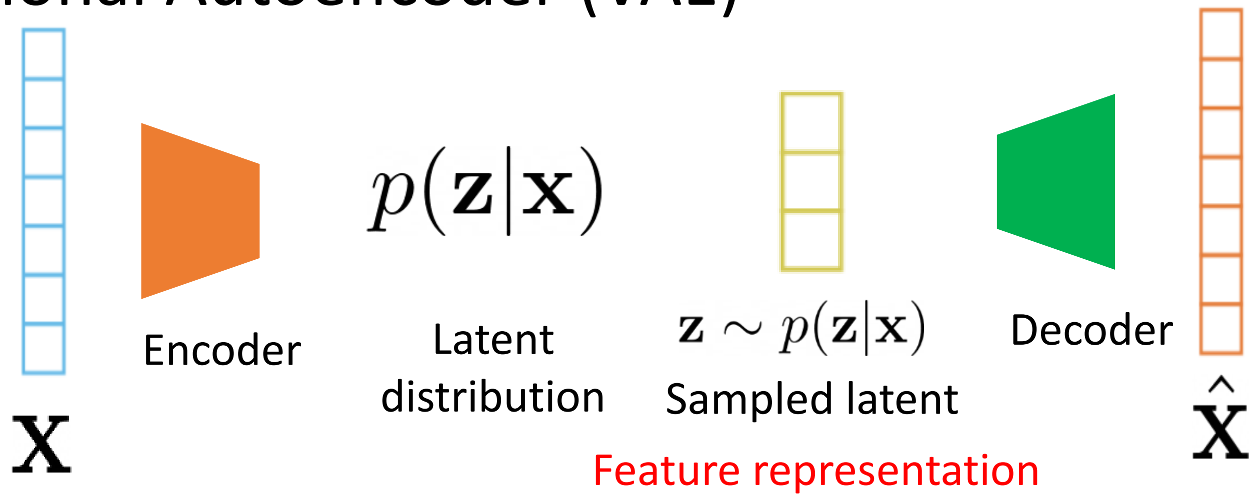
Classification
Clustering
Segmentation
Detection
Image captioning
Etc.

Generative Models

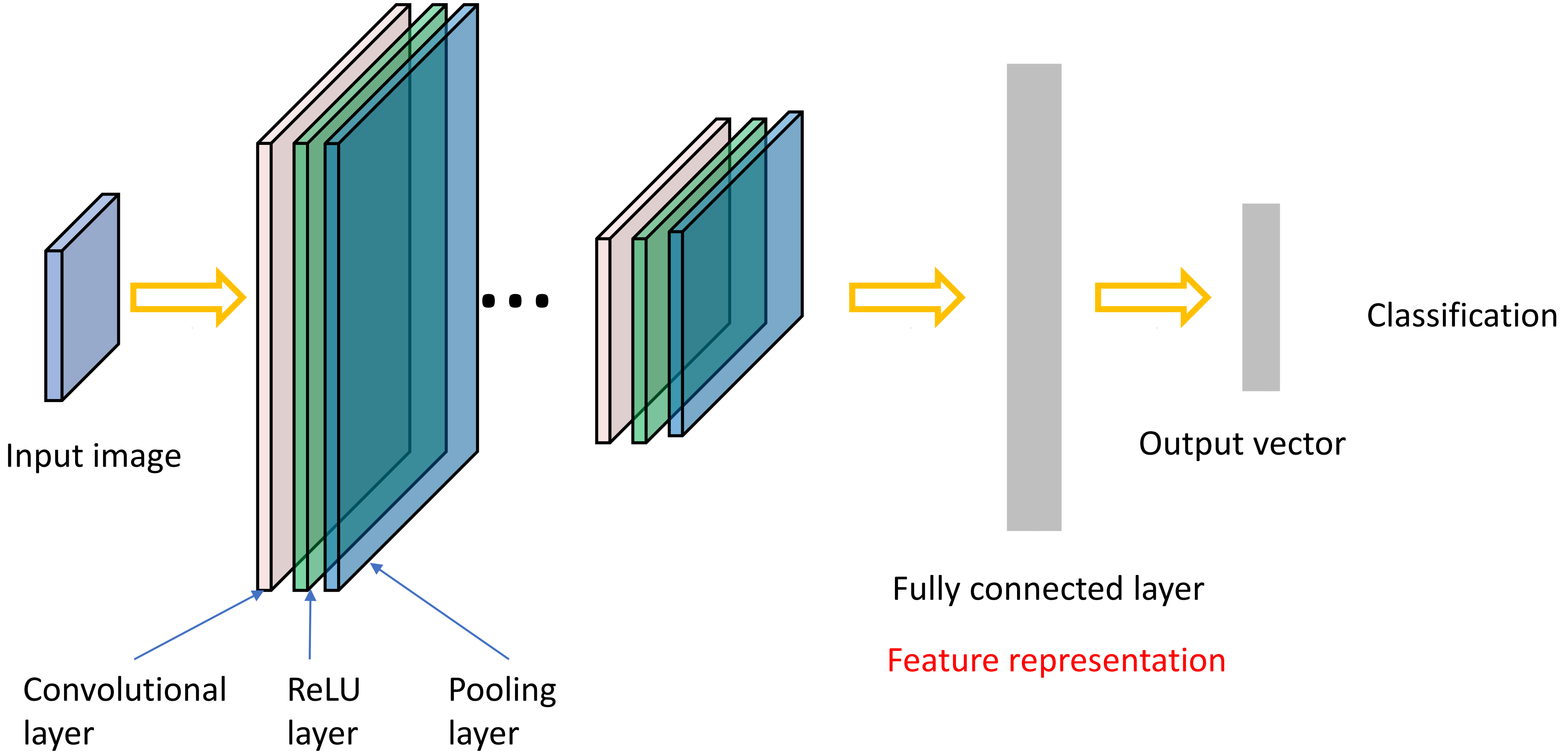
- Autoencoder



- Variational Autoencoder (VAE)



Discriminative Models (Supervised Learning)



Supervised Representation Learning

- Train neural networks for image classification
- Use internal features in the network as feature representations
- Applications

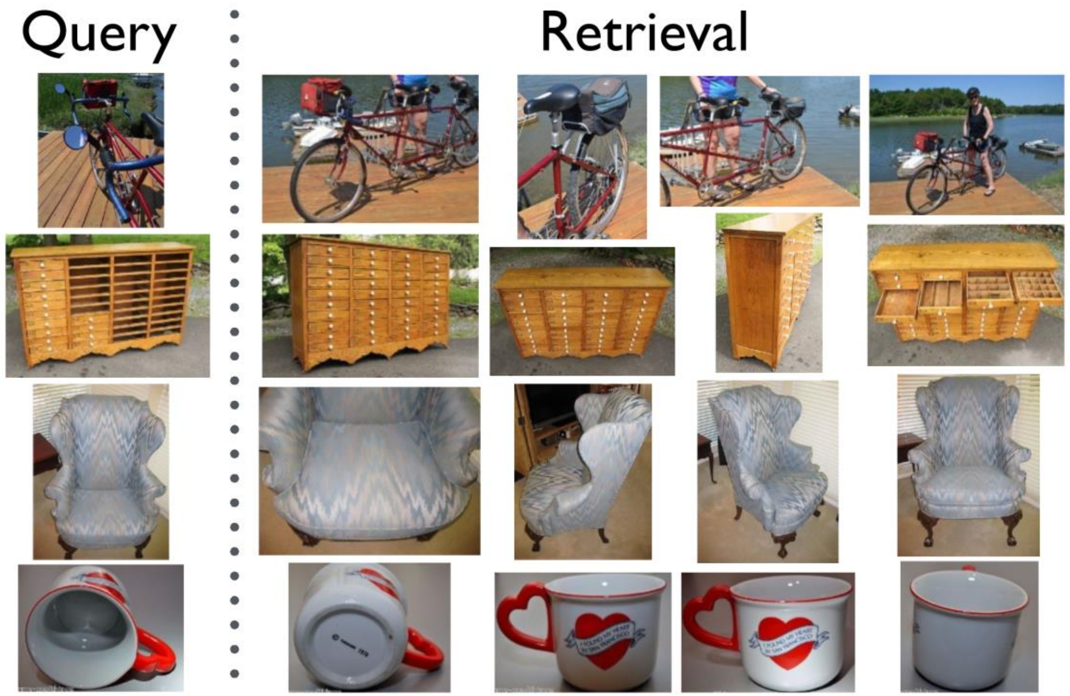
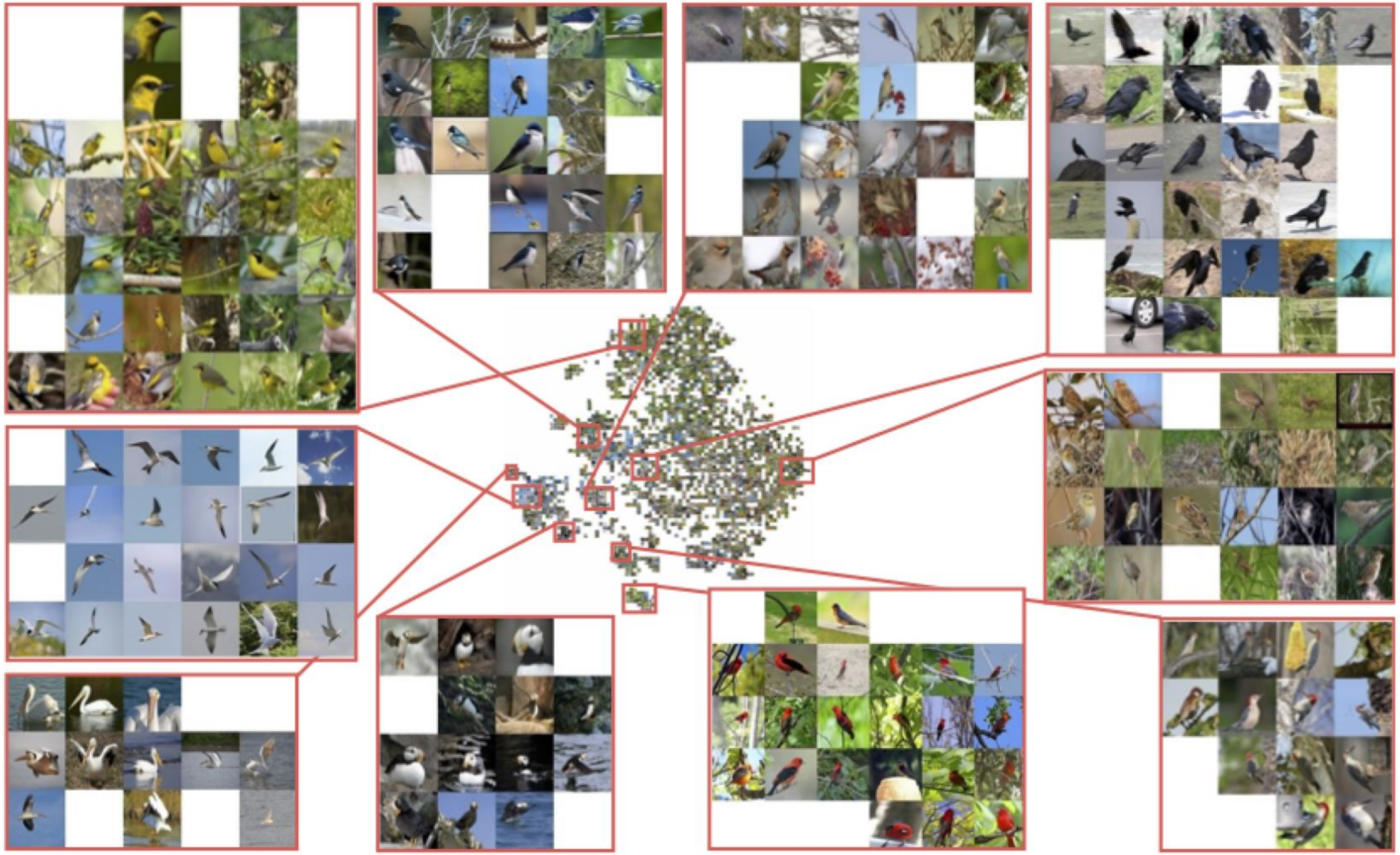


Image retrieval

Deep Metric Learning via Lifted Structured Feature Embedding. Song et al., CVPR, 2016.

Supervised Representation Learning

Image clustering



t-Distributed Stochastic Neighbor Embedding (t-SNE)

L.J.P. van der Maaten and G.E. Hinton. **Visualizing High-Dimensional Data Using t-SNE**. *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008.

Deep Metric Learning via Lifted Structured Feature Embedding. Song et al., CVPR, 2016.

Supervised Representation Learning

- Training with classification loss functions
  - E.g., cross-entropy loss

- Can we have better loss functions for representation learning?

- Deep metric learning
  - Learning distance metrics with neural networks

Distance metrics

- L1 distance

$$D(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} |x_i - y_i|$$

- L2 distance

$$D(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{N} (x_i - y_i)^2}$$

- Cosine distance

$$D(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\| \|\mathbf{y}\|}$$

Cosine similarity

Deep Metric Learning

$$\mathbf{x} \longrightarrow \boxed{\text{Neural Network}} \longrightarrow f(\mathbf{x})$$

Feature representation

$$D(\mathbf{x}_1, \mathbf{x}_2) = D(f(\mathbf{x}_1), f(\mathbf{x}_2))$$

L2 distance $$D(\mathbf{x}_1, \mathbf{x}_2) = \|f(\mathbf{x}_1) - f(\mathbf{x}_2)\|_2$$

Learning the distance metric is equivalent to learning the feature representation

Contrastive Loss

- Use positive pairs and negative pairs

$\mathbf{x}_1 \rightarrow$ Neural Network $\rightarrow f(\mathbf{x}_1)$

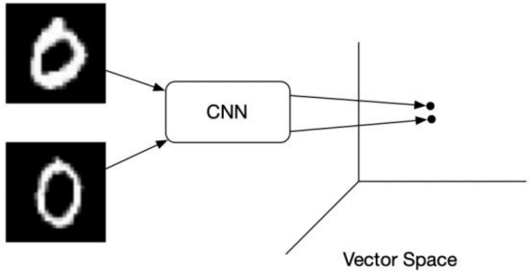
$\mathbf{x}_2 \rightarrow$ Neural Network $\rightarrow f(\mathbf{x}_2)$

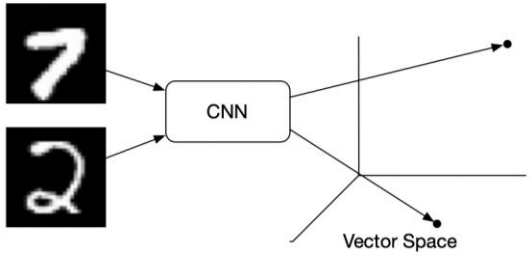
Positive pair $f(\mathbf{x}_1)$ $f(\mathbf{x}_2)$ should be close

$D(\mathbf{x}_1, \mathbf{x}_2)$ small

Negative pair $f(\mathbf{x}_1)$ $f(\mathbf{x}_2)$ should be far

$D(\mathbf{x}_1, \mathbf{x}_2)$ large





Learning a Similarity Metric Discriminatively, with Application to Face Verification. Chopra et al., CVPR, 2005.

Contrastive Loss

- Training data $\{(\mathbf{x}_i, \mathbf{x}_j, y_{ij})\}$ $\quad y_{ij} = \begin{cases} 1 & \text{if positive pair} \\ 0 & \text{if negative pair} \end{cases}$

(a) Contrastive embedding

$$J = \frac{1}{m} \sum_{(i,j)}^{m/2} y_{i,j} D_{i,j}^2 + (1 - y_{i,j}) \left[\alpha - D_{i,j}\right]_+^2$$

margin

$$[x]_+ = \max(0, x)$$

m: number of images in a batch

Learning a Similarity Metric Discriminatively, with Application to Face Verification. Chopra et al., CVPR, 2005.

Contrastive Loss

- Compute Gradient

$$J = \frac{1}{m} \sum_{(i,j)}^{m/2} y_{i,j} D_{i,j}^2 + (1 - y_{i,j}) \left[\alpha - D_{i,j}\right]_+^2$$

$$\frac{\partial J}{\partial D_{i,j}} = \frac{2}{m}\left(y_{i,j} D_{i,j} - (1 - y_{i,j})[\alpha - D_{i,j}]_+\right)$$

$$D_{i,j} = \|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|_2 \qquad \frac{\partial D_{i,j}}{\partial f(\mathbf{x}_i)} = \frac{f(\mathbf{x}_i) - f(\mathbf{x}_j)}{\|f(\mathbf{x}_i) - f(\mathbf{x}_j)\|}$$

$\mathbf{x}_i \rightarrow$ Neural Network $\rightarrow f(\mathbf{x}_i)$

$\leftarrow$ Gradients $\leftarrow$ $\frac{\partial J}{\partial f(\mathbf{x}_i)}$

Triplet Loss

- Use a triplet (anchor, positive, negative)

(b) Triplet embedding

$$J = \frac{3}{2m} \sum_{i}^{m/3} \left[D_{ia,ip}^2 - D_{ia,in}^2 + \alpha \right]_+$$

$$D_{ia,ip} = ||f(\mathbf{x}_i^a) - f(\mathbf{x}_i^p)|| \qquad D_{ia,in} = ||f(\mathbf{x}_i^a) - f(\mathbf{x}_i^n)||$$

FaceNet: A Unified Embedding for Face Recognition and Clustering. Schroff et al., CVPR, 2015.

Lifted Structured Loss

- Consider all positive pairs and negative pairs in a mini-batch

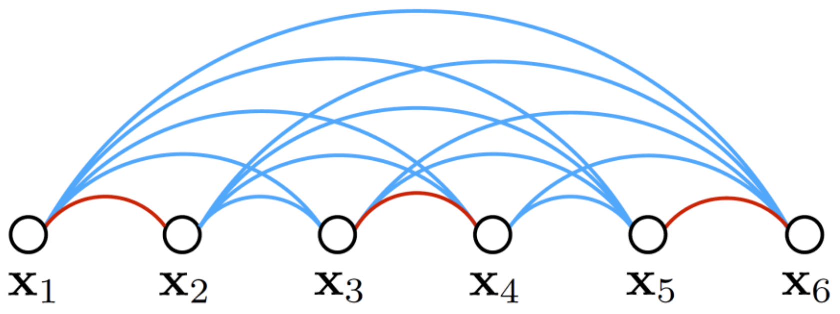
$$J = \frac{1}{2|\widehat{\mathcal{P}}|} \sum_{(i,j)\in\widehat{\mathcal{P}}} \max\left(0,\ J_{i,j}\right)^2$$

$$J_{i,j} = \max\left(\max_{(i,k)\in\widehat{\mathcal{N}}} \alpha - D_{i,k},\ \max_{(j,l)\in\widehat{\mathcal{N}}} \alpha - D_{j,l} \right) + D_{i,j}$$

Hard negative

Distance for the negative pair

Distance for the positive pair



(c) Lifted structured embedding

Relaxed loss $$\tilde{J}_{i,j} = \log\left(\sum_{(i,k)\in\mathcal{N}} \exp\{\alpha - D_{i,k}\} + \sum_{(j,l)\in\mathcal{N}} \exp\{\alpha - D_{j,l}\} \right) + D_{i,j}$$

Deep Metric Learning via Lifted Structured Feature Embedding. Song et al., CVPR, 2016.

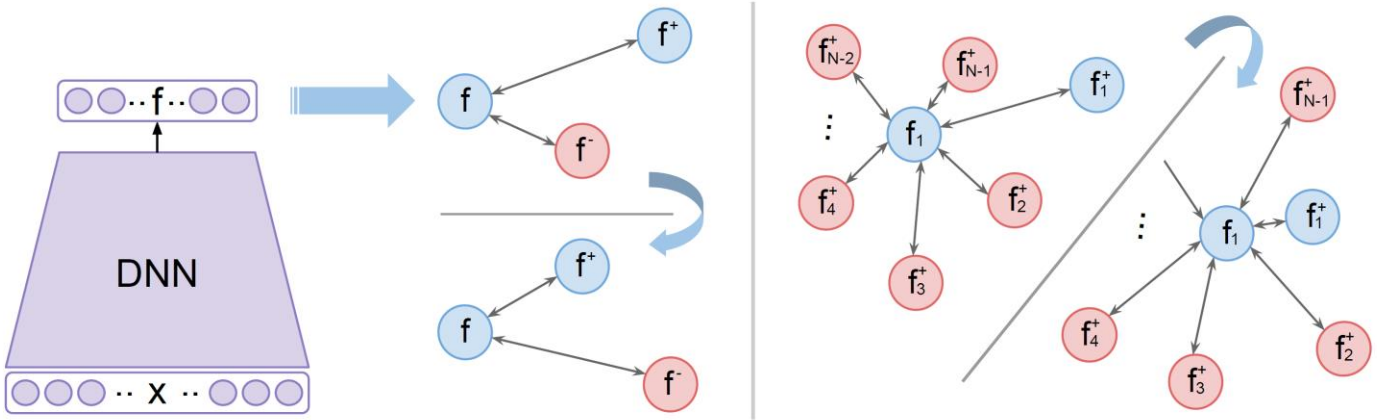
Multi-class N-pair Loss

- Use a positive pair and N-1 negative ones and $\{\mathbf{x}, \mathbf{x}^+, \mathbf{x}_1^-, \ldots, \mathbf{x}_{N-1}^-\}$.

$$\mathcal{L}_{\text{N-pair}}(\mathbf{x}, \mathbf{x}^+, \{\mathbf{x}_i^-\}_{i=1}^{N-1}) = \log\left(1 + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-) - f(\mathbf{x})^\top f(\mathbf{x}^+))\right)$$

$$= -\log \frac{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+))}{\exp(f(\mathbf{x})^\top f(\mathbf{x}^+)) + \sum_{i=1}^{N-1} \exp(f(\mathbf{x})^\top f(\mathbf{x}_i^-))}$$

Softmax for multi-class classification



Improved Deep Metric Learning with Multi-class N-pair Loss Objective. Kihyuk Sohn, NeurIPS, 2016

InfoNCE (Noise Contrastive Estimation) Loss

- Similar to multi-class N-pair Loss

$$\mathcal{L}_q = -\log \frac{\exp(q \cdot k_+ / \tau)}{\sum_{i=0}^{K} \exp(q \cdot k_i / \tau)}$$

Query q

Positive k+

Negatives ki

(K+1)-way softmax classification

Motivated from identifying targets from noisy data

Supervised Representation Learning

- Use class labels to specify positive pairs and negative pairs

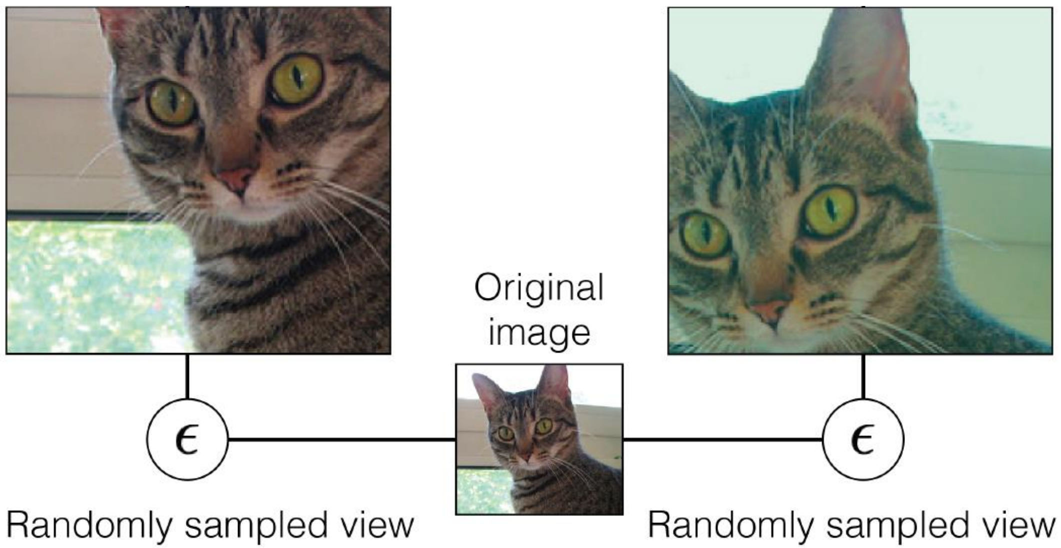
- Loss functions
  - Contrastive loss
  - Triplet loss
  - Lifted structured loss
  - N-pair loss
  - InfoNCE

- Consider more relationships in a mini-batch is better

Unsupervised/Self-supervised Representation Learning

- Pretext tasks
  - Tasks designed for feature learning
  - Not the final tasks

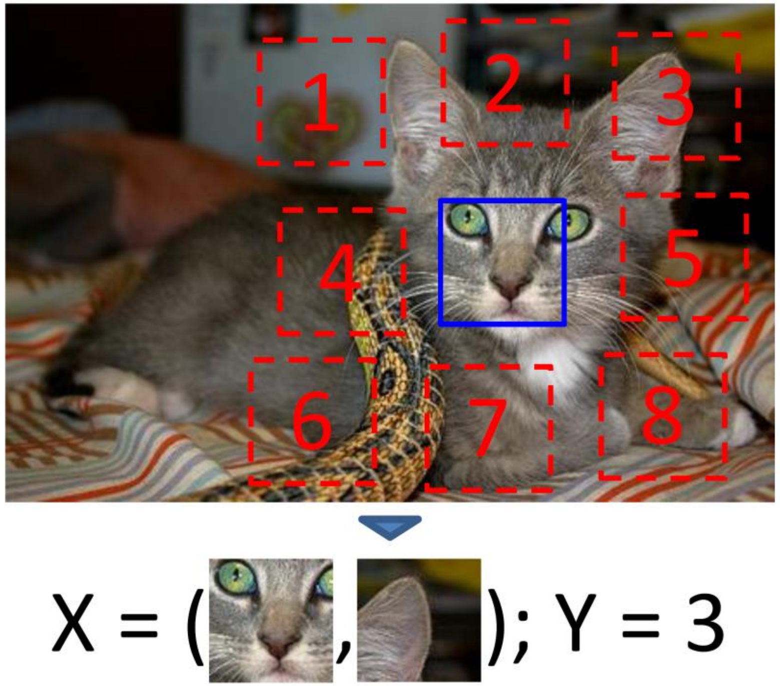
- Positive pairs from different views of the same image

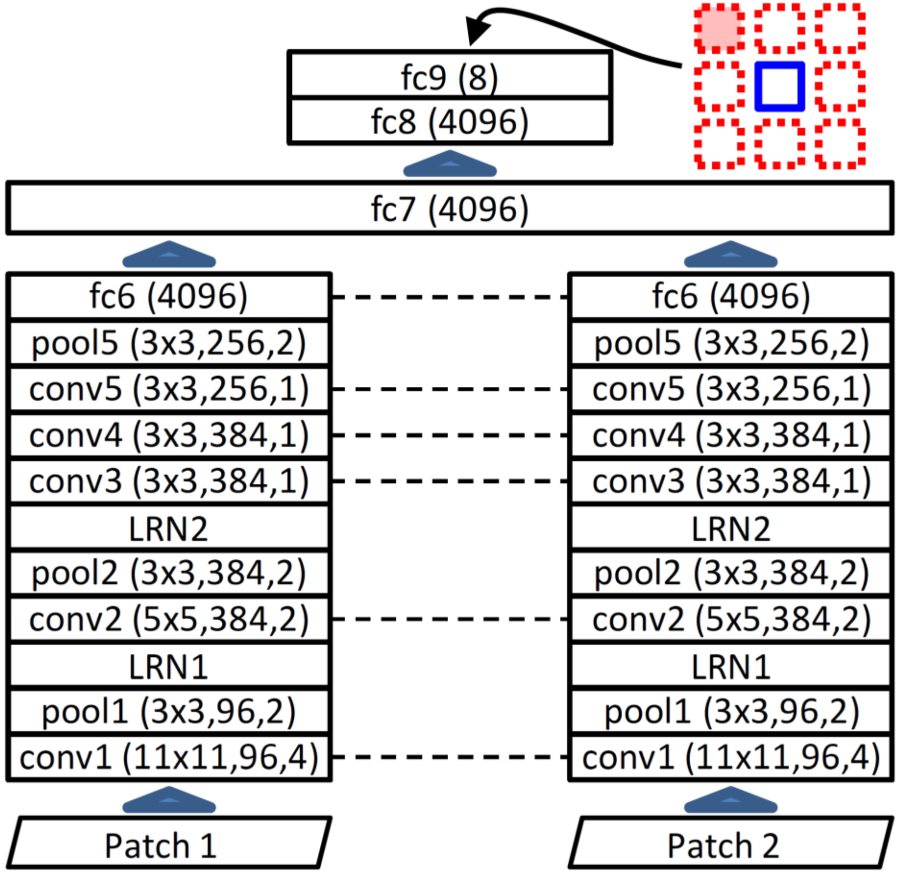


Learning Representations by Maximizing Mutual Information Across Views. Bachman et al., NeurIPS, 2019

Unsupervised/Self-supervised Representation Learning

- Pretext task: context prediction



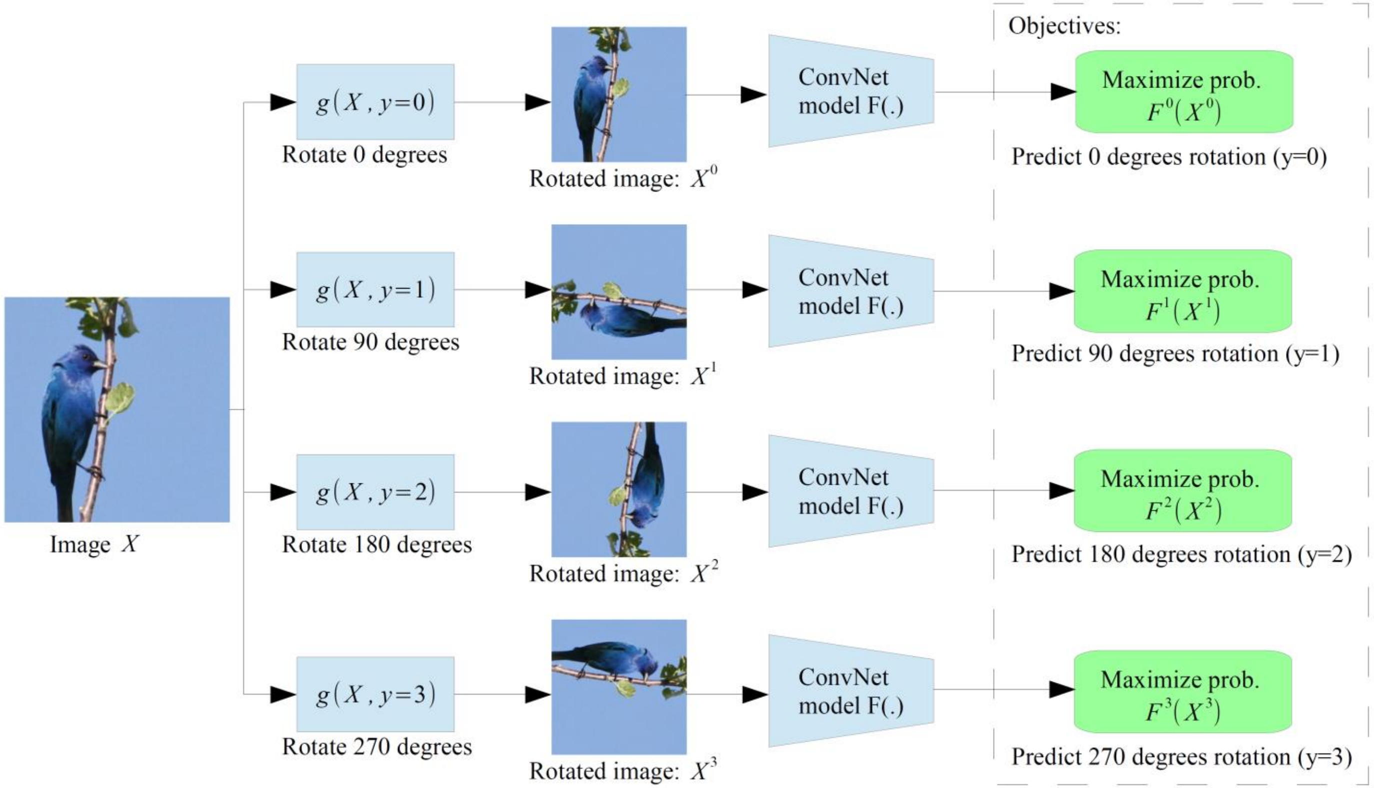


Feature representation

Unsupervised Visual Representation Learning by Context Prediction. Doersch, et al., ICCV, 2015

Unsupervised/Self-supervised Representation Learning

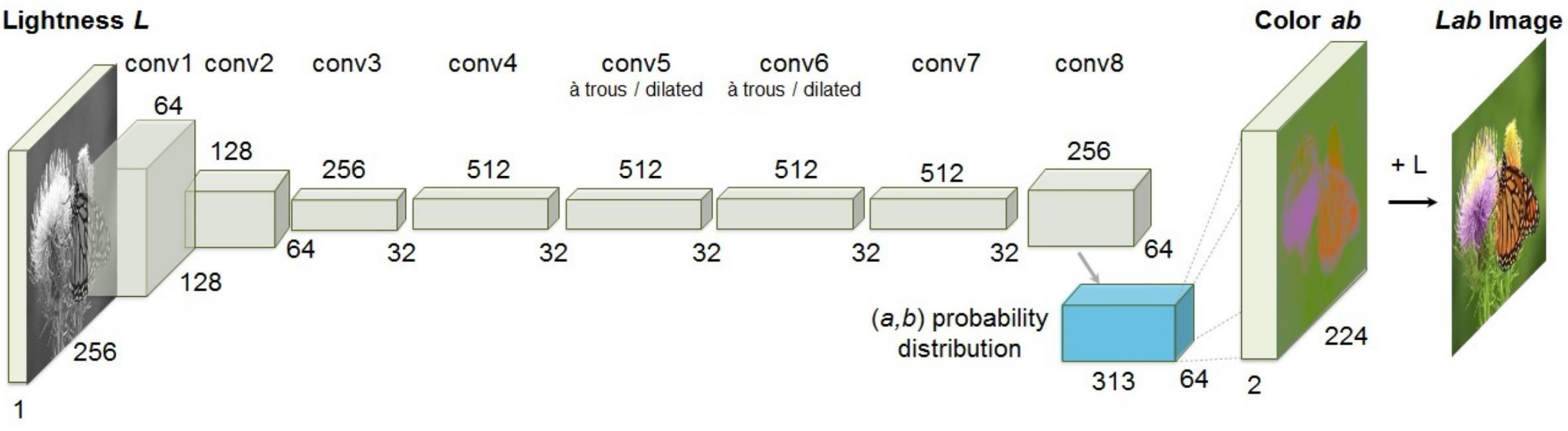
- Pretext task: rotation prediction



Unsupervised Representation Learning by Predicting Image Rotations. Gidaris, et al., ICLR, 2018

Unsupervised/Self-supervised Representation Learning

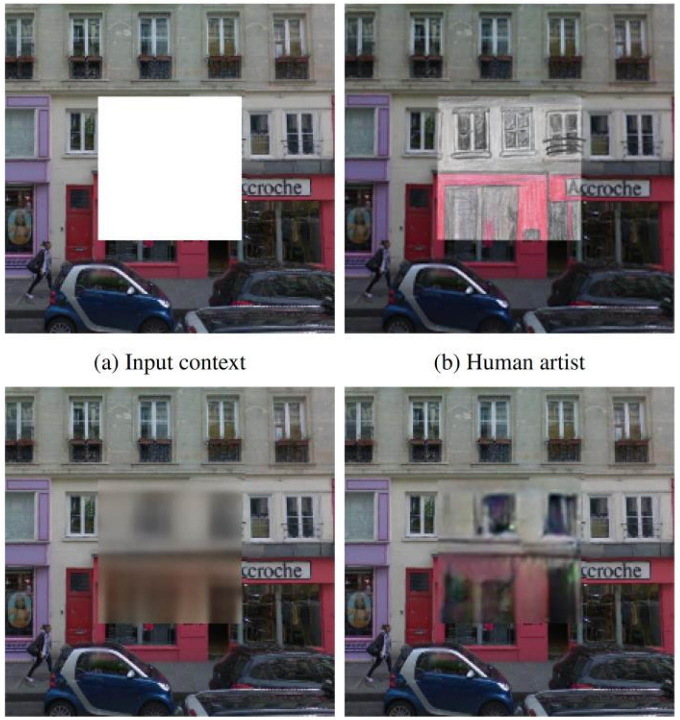
- Pretext task: colorization



Colorful Image Colorization. Zhang, et al., ECCV, 2016

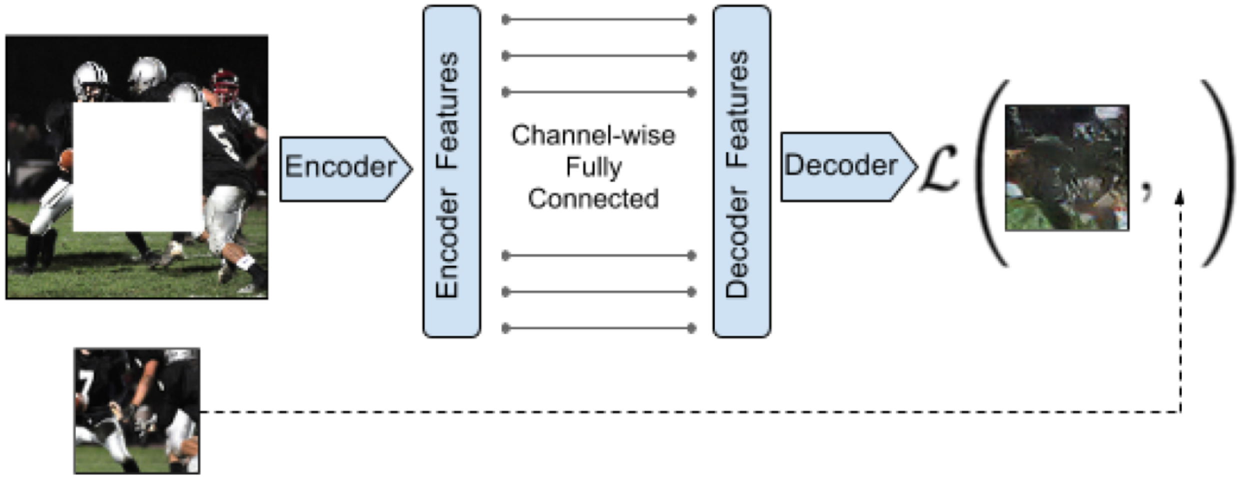
Unsupervised/Self-supervised Representation Learning

- Pretext task: inpainting



(a) Input context (b) Human artist

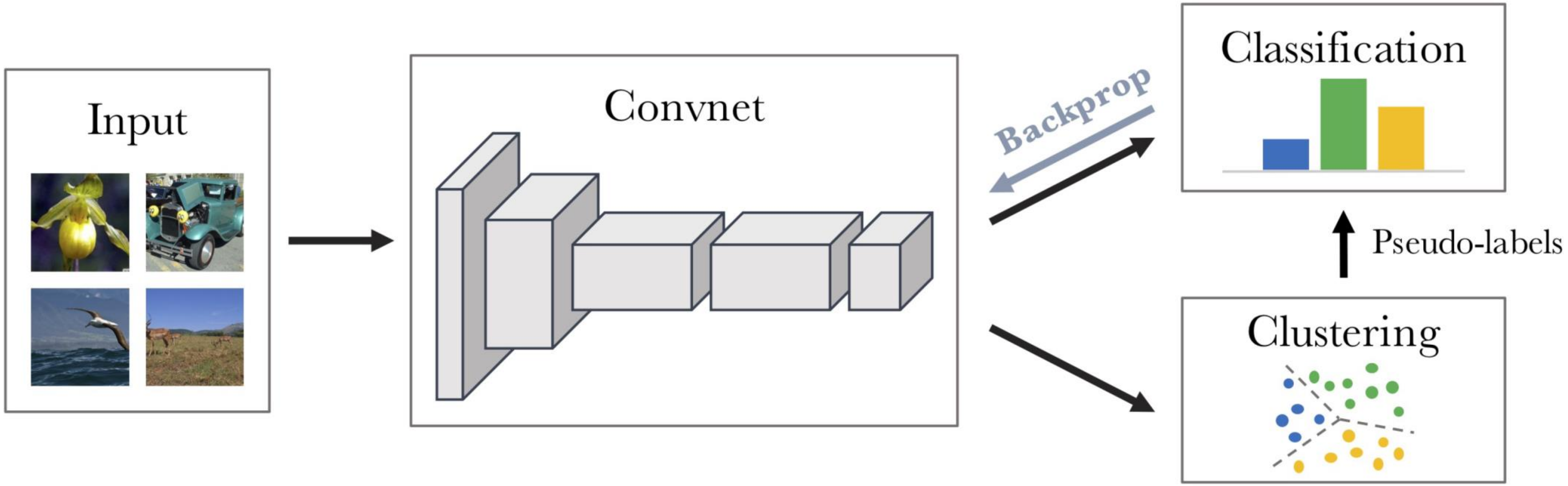
(c) Context Encoder ($L2$ loss) (d) Context Encoder ($L2$ + Adversarial loss)



Context Encoders: Feature Learning by Inpainting. Pathak, et al., CVPR, 2016

Unsupervised/Self-supervised Representation Learning

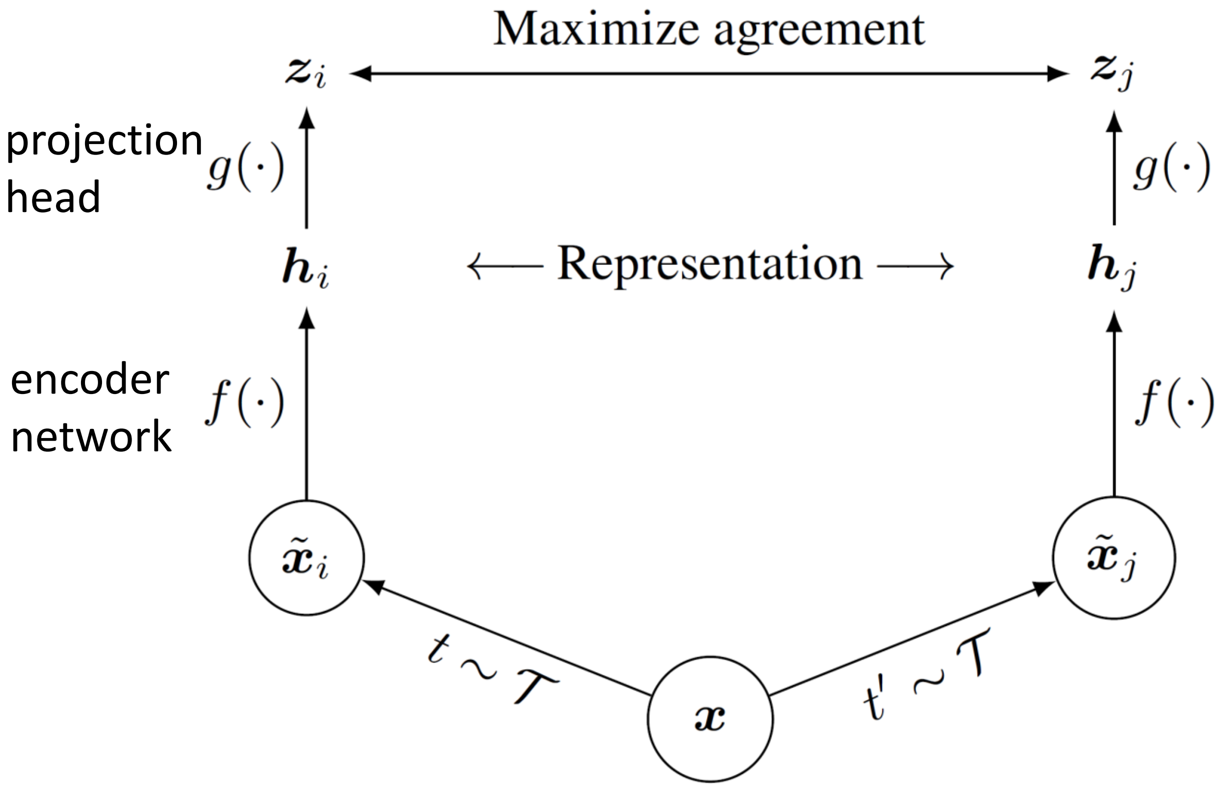
- Pretext task: clustering



Deep Clustering for Unsupervised Learning of Visual Features. Caron et al., ECCV, 2018

SimCLR

- A simple framework for contrastive learning of visual representations



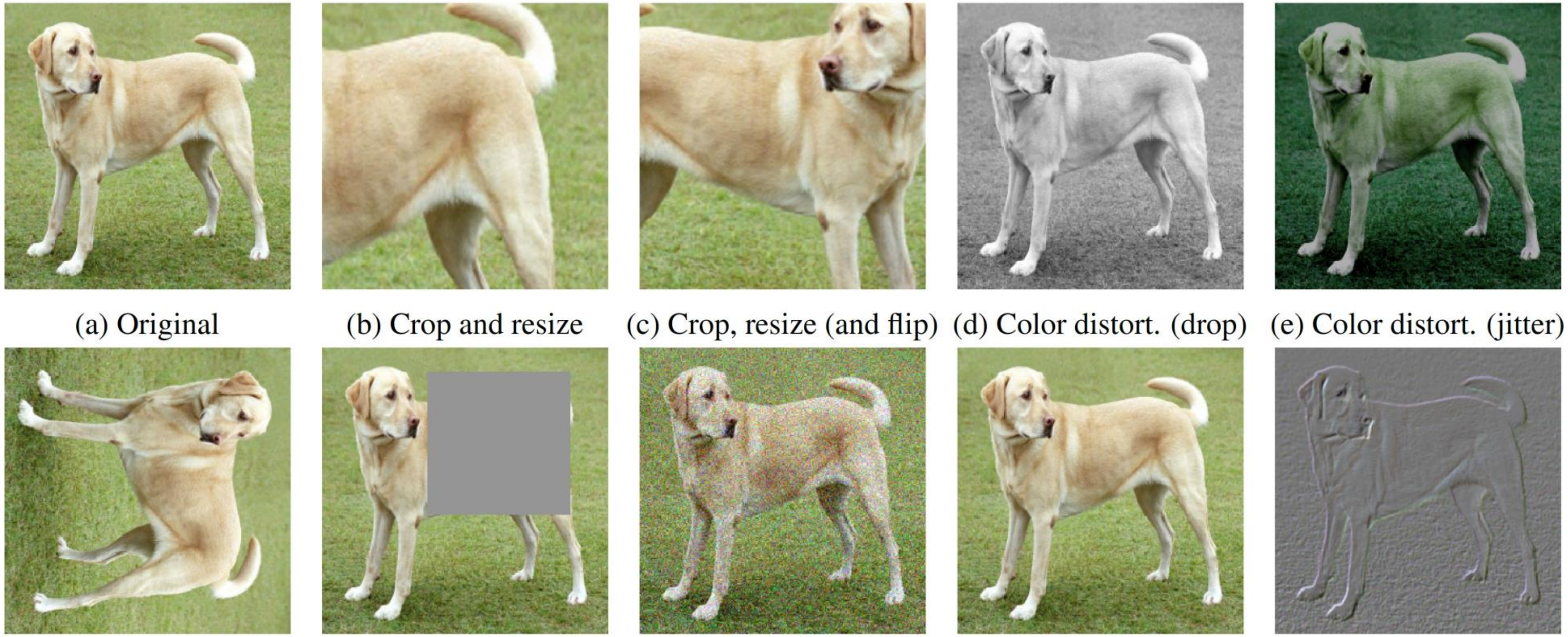
Loss function

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(z_i, z_k)/\tau)}$$

A Simple Framework for Contrastive Learning of Visual Representations. Chen et al., ICML, 2020

SimCLR

- Transformations



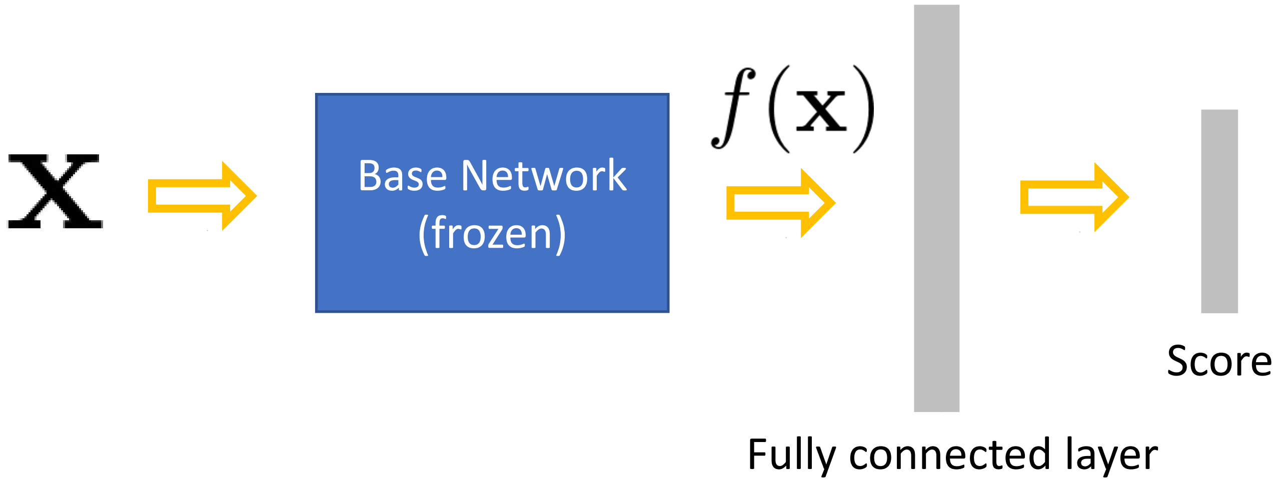
(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

(f) Rotate $\{90°, 180°, 270°\}$ (g) Cutout (h) Gaussian noise (i) Gaussian blur (j) Sobel filtering

A Simple Framework for Contrastive Learning of Visual Representations. Chen et al., ICML, 2020

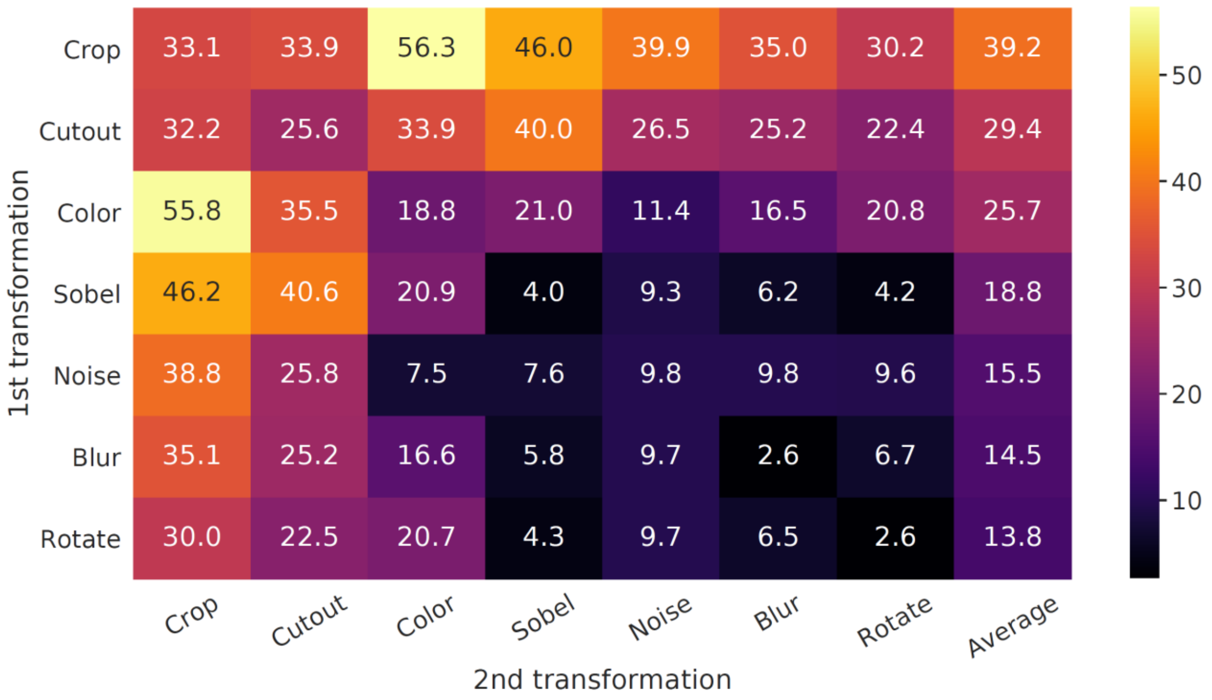
SimCLR

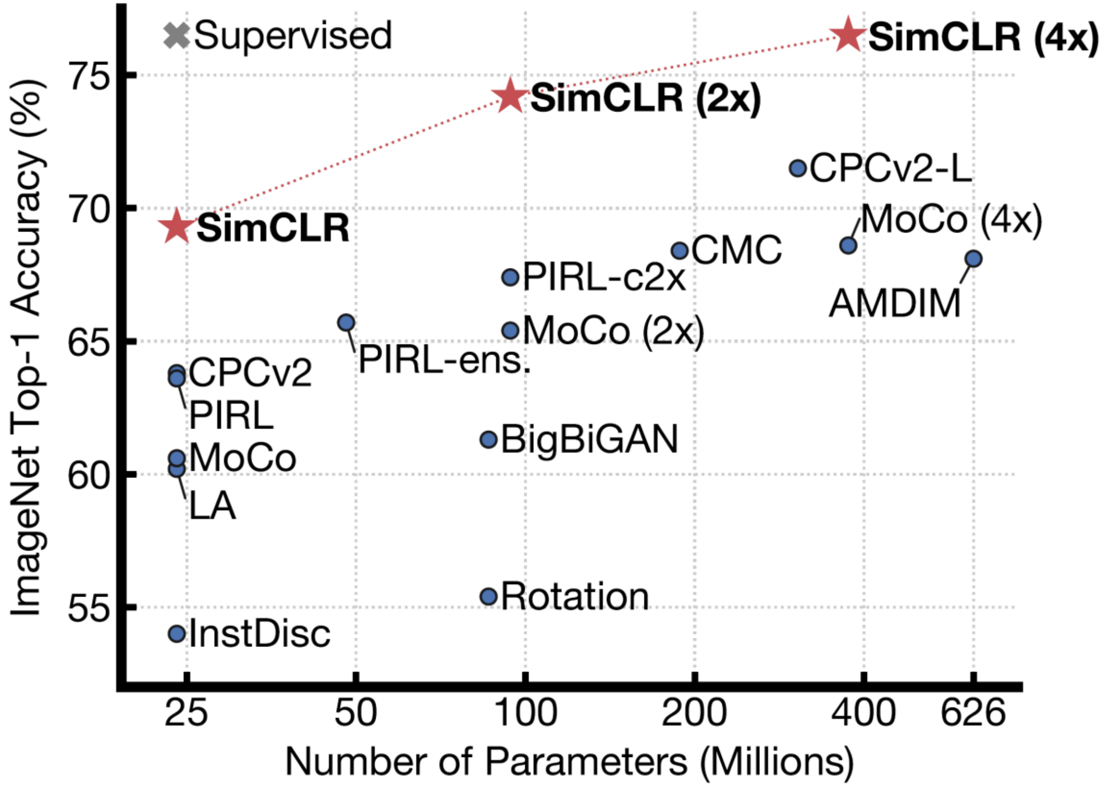
- After training, keep the encoder network $h_i = f(\tilde{x}_i) = \mathrm{ResNet}(\tilde{x}_i)$
- Linear evaluation protocol for classification
  - A linear classifier is trained on top of the frozen base network



A Simple Framework for Contrastive Learning of Visual Representations. Chen et al., ICML, 2020

SimCLR





ImageNet top-1 accuracy

2x, 4x: more channels in ResNet

A Simple Framework for Contrastive Learning of Visual Representations. Chen et al., ICML, 2020

SimCLR

https://github.com/google-research/simclr

Summary: Visual Representation Learning

- Generative models
    - Autoencoder
    - VAE

- Discriminative models
    - Supervised learning
        - Training with image classification
        - Deep metric learning
    - Unsupervised/self-supervised learning
        - Use pretext tasks
        - Metric learning loss functions

Further Reading

- Learning a Similarity Metric Discriminatively, with Application to Face Verification, 2005 http://yann.lecun.com/exdb/publis/pdf/chopra-05.pdf
- FaceNet: A Unified Embedding for Face Recognition and Clustering, 2015 https://arxiv.org/abs/1503.03832
- Deep Metric Learning via Lifted Structured Feature Embedding, 2016 https://arxiv.org/abs/1511.06452
- Improved Deep Metric Learning with Multi-class N-pair Loss Objective, 2016 https://papers.nips.cc/paper/2016/file/6b180037abbebea991d8b1232f8a8ca9-Paper.pdf
- Learning Representations by Maximizing Mutual Information Across Views, 2019 https://arxiv.org/pdf/1906.00910.pdf
- A Simple Framework for Contrastive Learning of Visual Representations, 2020 https://arxiv.org/abs/2002.05709