SELF-SUPERVISED LEARNING

Introduction

- Supervised learning – learning with labeled data

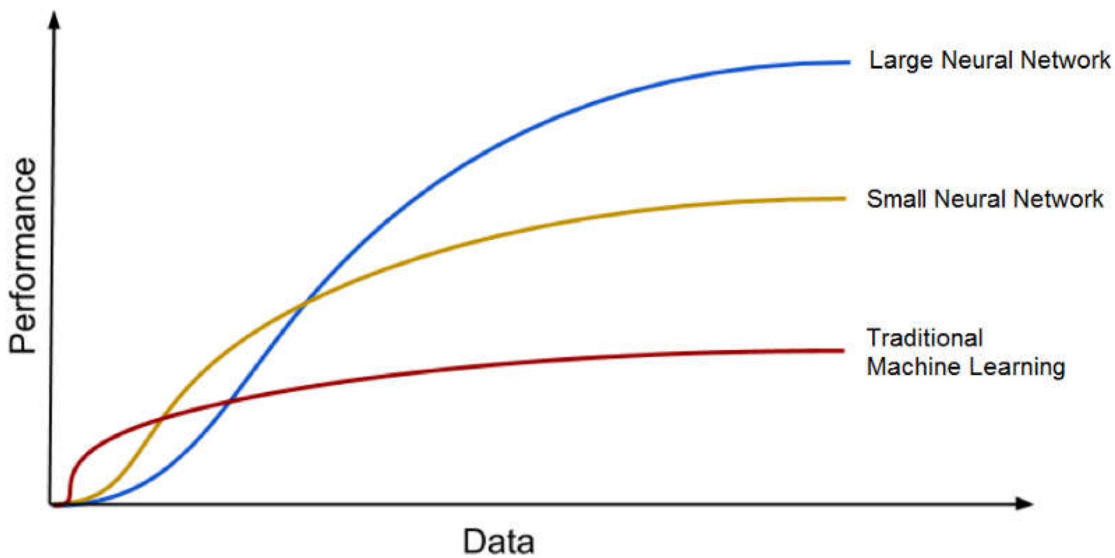
  Collect a dataset with labels (labels are expensive)

- Unsupervised learning – learning with unlabeled data

  Collect a large dataset without label (unlabeled data are cheap)

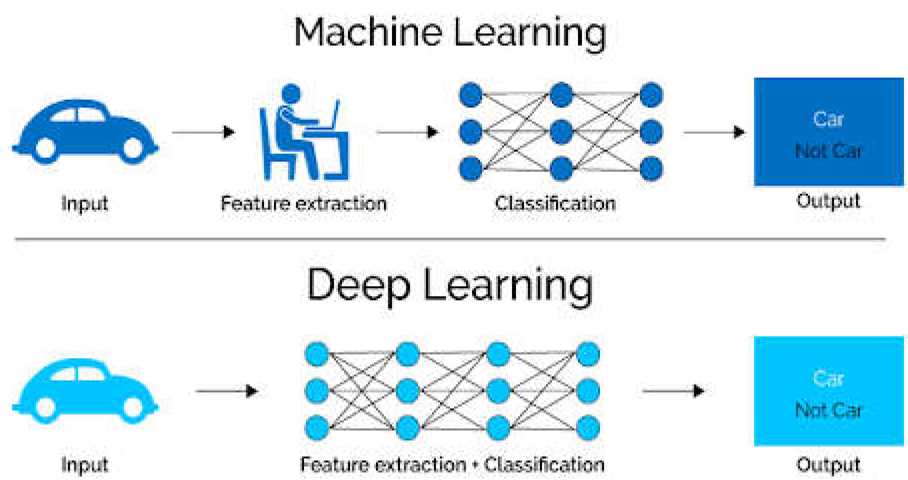
Deep Neural Networks: Very large Models (many parameters)
How to train?

Introduction

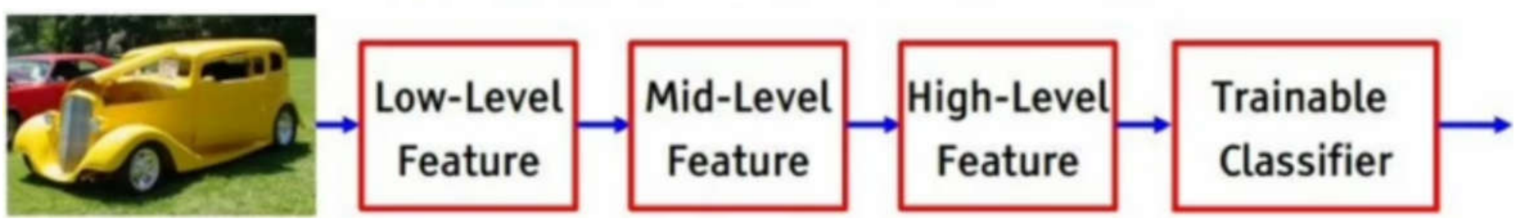


https://builtin.com/artificial-intelligence/ai-vs-machine-learning

Introduction



$$y = f_L(\dots f_3(f_2(f_1(x|\theta_1)|\theta_2)|\theta_3)\dots|\theta_L)$$



TRANSFER LEARNING

Transfer Learning

- knowledge of an already trained machine learning model is applied to a different but related problem
- The general idea is to use the knowledge a model has learned from a task with a lot of available labeled training data in a new task that doesn't have much data.
- it has become quite popular in combination with neural networks that require huge amounts of data and computational power.

Transfer Learning

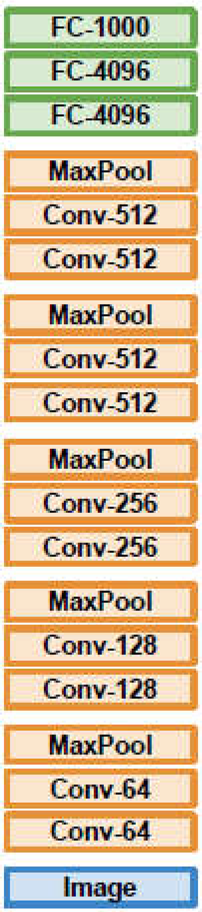
- In computer vision, neural networks usually try to detect edges in the earlier layers, shapes in the middle layer and some task-specific features in the later layers.

$$y = f_L(\ldots f_3(f_2(f_1(x|\theta_1)|\theta_2)|\theta_3) \ldots |\theta_L)$$

- In transfer learning, the early and middle layers are used
- only retrain the latter layers
- saving training time
- good performance of neural networks (in most cases)
- not needing a lot of data.

Transfer Learning

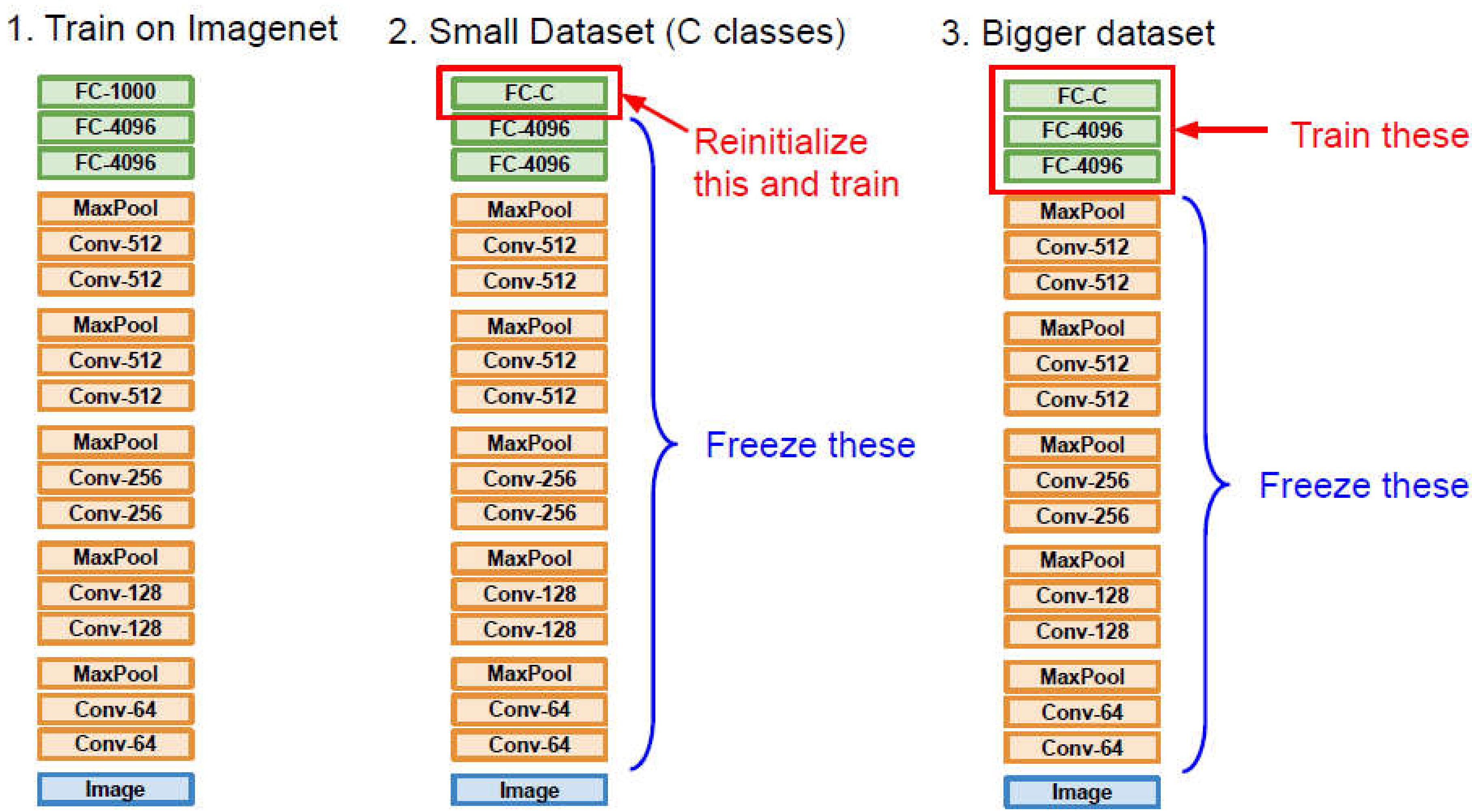
1. Train on Imagenet



ImageNet

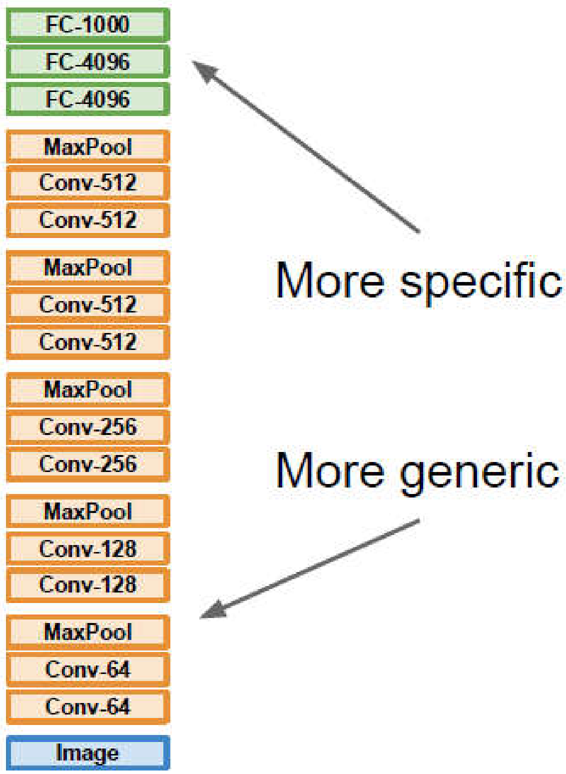
Lectures of deep learning for computer vision course (CS231n course -Stanford university)

Transfer Learning



Lectures of deep learning for computer vision course (CS231n course -Stanford university)

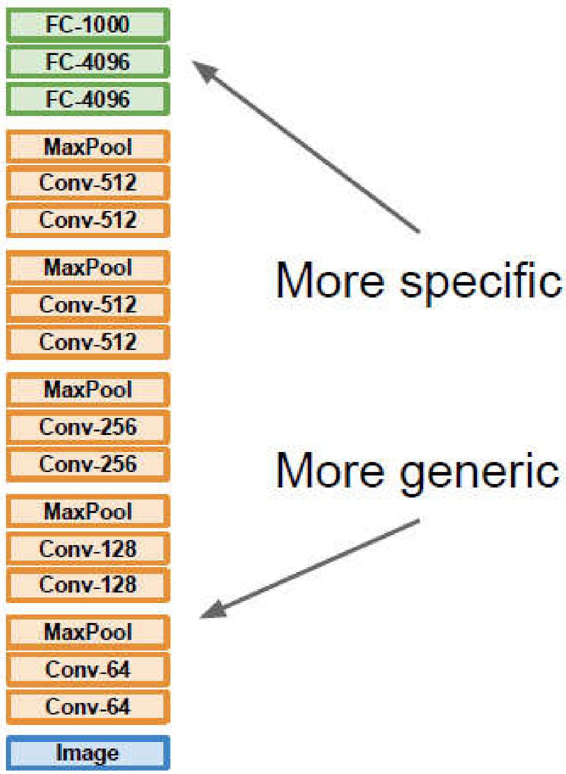
Transfer Learning



	very similar dataset	very different dataset
very little data	?	?
quite a lot of data	?	?

Lectures of deep learning for computer vision course (CS231n course -Stanford university)

Transfer Learning



	very similar dataset	**very different dataset**
very little data	Use Linear Classifier on top layer	You're in trouble… Try linear classifier from different stages
quite a lot of data	Finetune a few layers	Finetune a larger number of layers

Lectures of deep learning for computer vision course (CS231n course -Stanford university)

SELF-SUPERVISED LEARNING

Self-supervised learning

Why self-supervised learning?

- ❖ Creating labeled datasets for each task is an expensive
- ❖ Vast amount of unlabeled data on the internet (images, videos, text)
- ❖ Extract good features

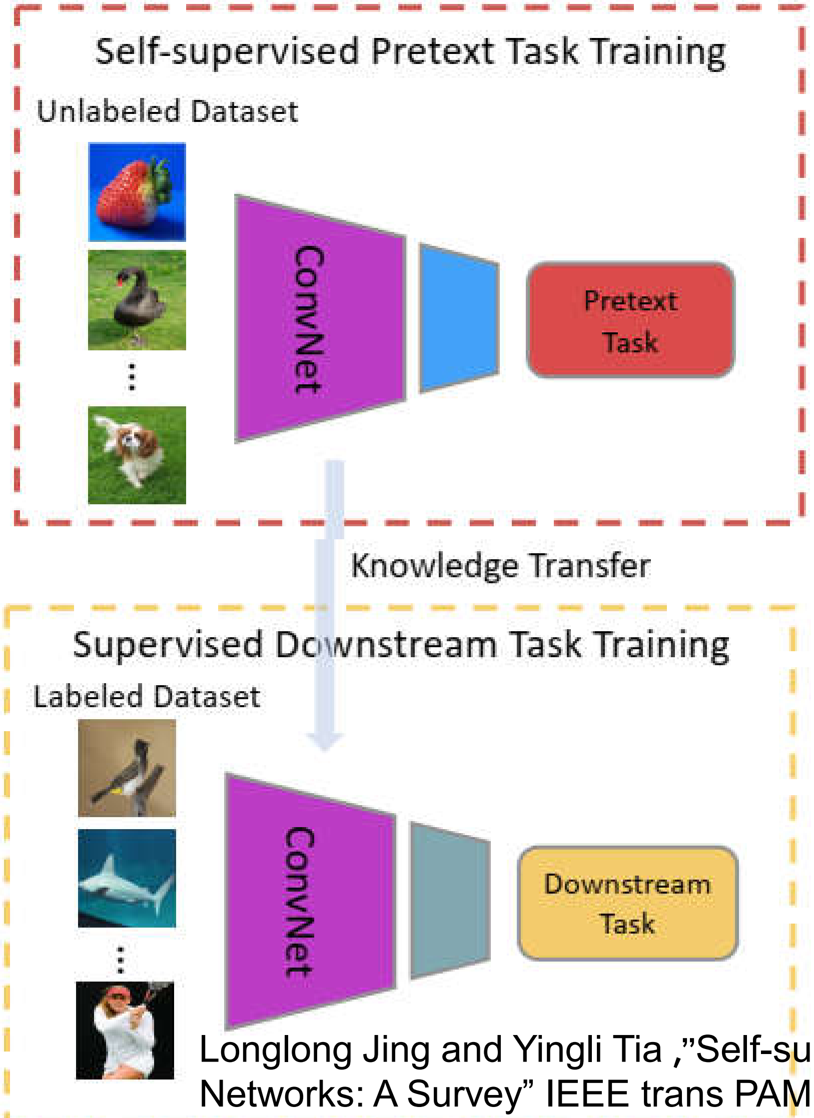
Self-supervised learning

- Supervised learning – learning with labeled data
- Unsupervised learning – learning with unlabeled data
- Self-supervised learning – a subclass of unsupervised learning

Goal: Learn useful representations through pretraining tasks for downstream tasks

$$y = f_L(\ldots f_3(f_2(f_1(x|\theta_1)|\theta_2)|\theta_3)\ldots|\theta_L)$$

Self-supervised learning



Pretext Task pre-designed tasks for networks to solve, and visual features are learned by learning objective functions of pretext tasks.

Downstream Task: applications that are used to evaluate the quality of features learned by self-supervised learning.

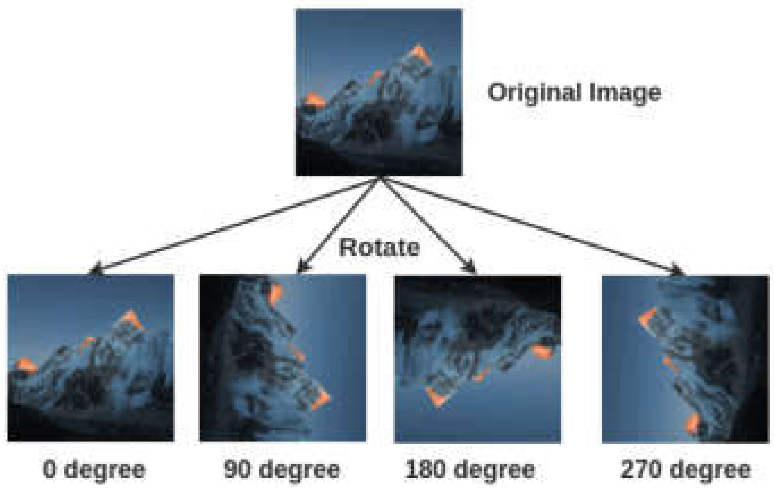
Pretext tasks:

- Not simple, sufficiently complex
- Pseudo label

Longlong Jing and Yingli Tia ,"Self-supervised Visual Feature Learning with Deep Neural Networks: A Survey" IEEE trans PAMI, 2020

Pretraining Tasks: Image rotation

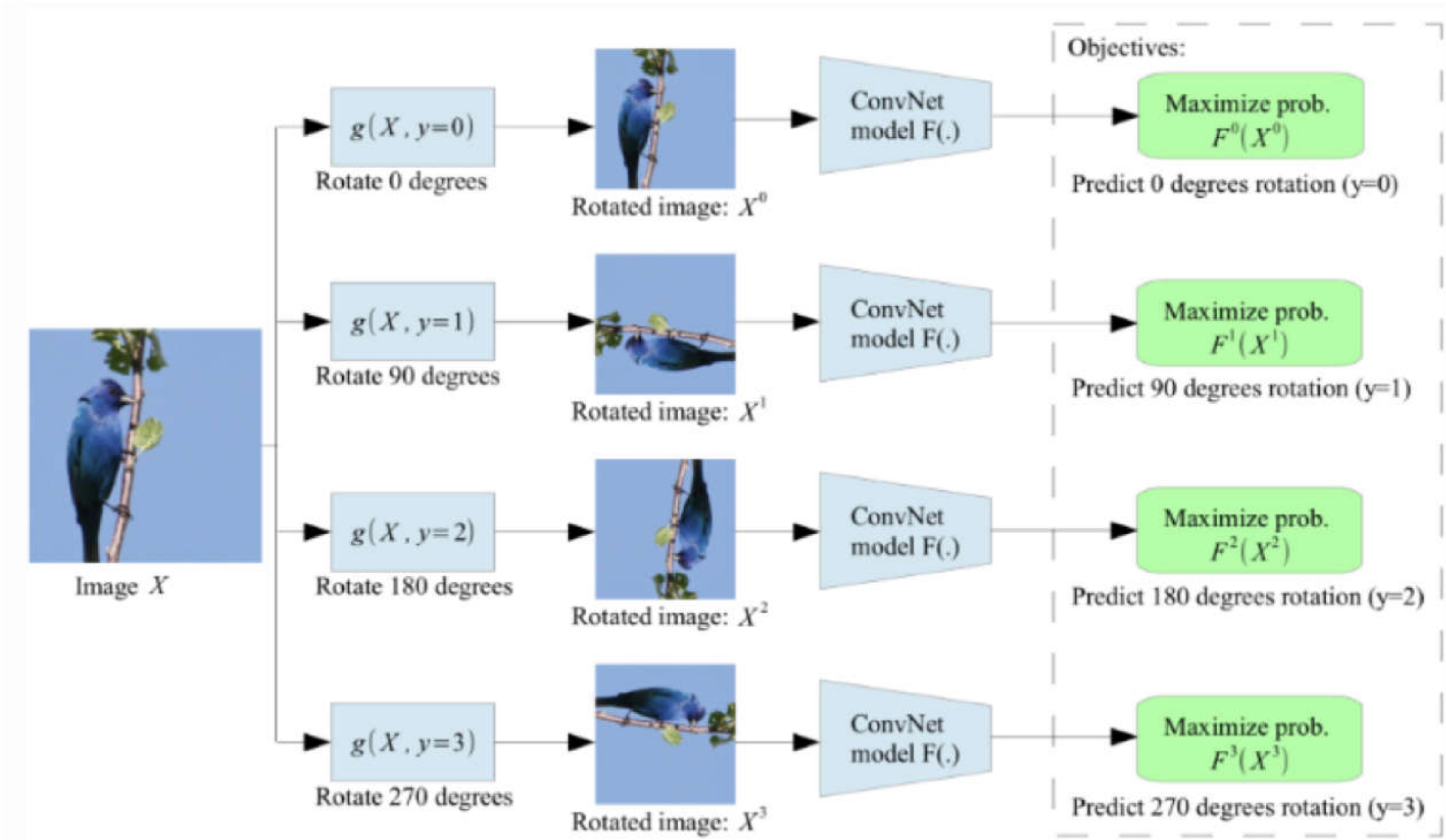
Geometric transformation recognition: Image rotation



Pretraining data

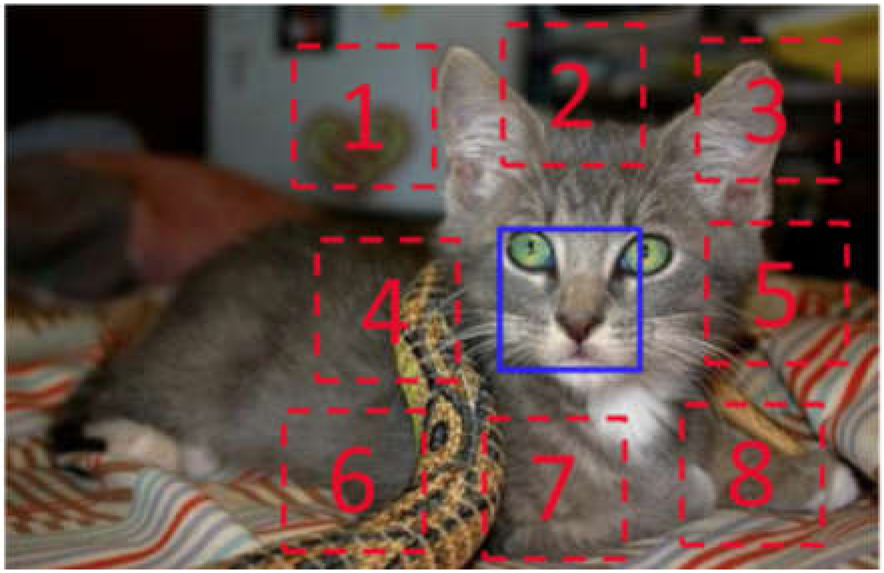
Gidaris (2018) - Unsupervised Representation Learning by Predicting Image Rotations

Pretraining Tasks: Image rotation



Gidaris (2018) - Unsupervised Representation Learning by Predicting Image Rotations

Relative Patch Position

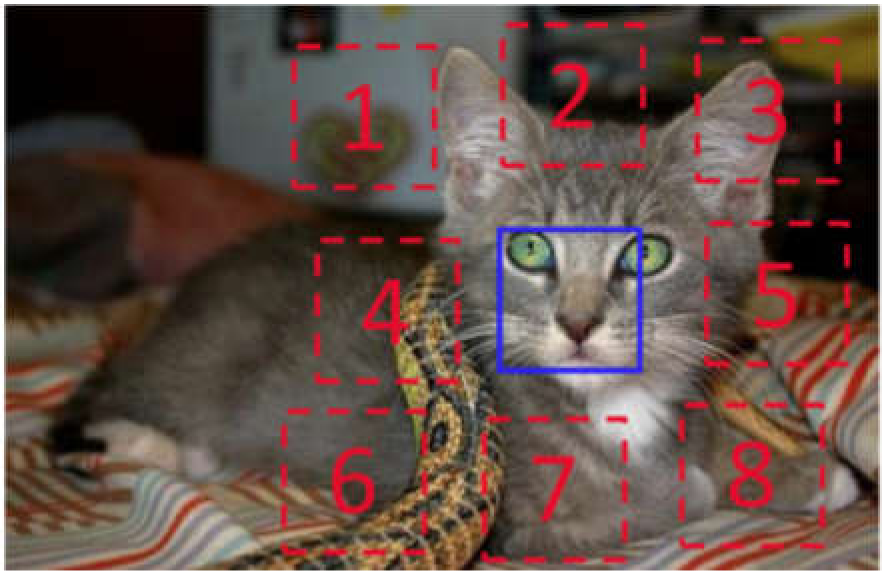


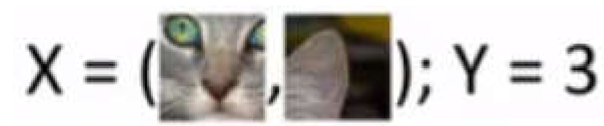
Pretraining data: multiple patches extracted from images

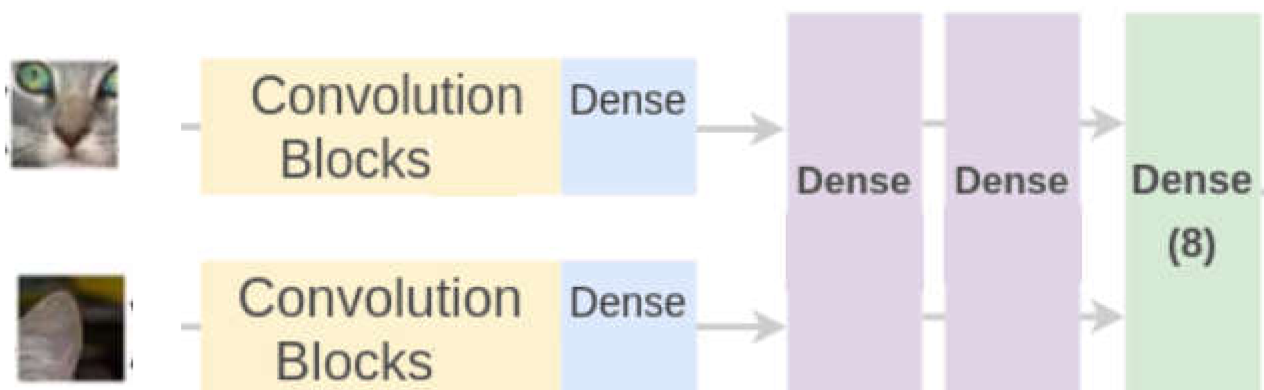
Pretraining task: train a model to predict the relationship between the patches

Dorsch (2015) Unsupervised Visual Representation Learning by Context Prediction

Relative Patch Position

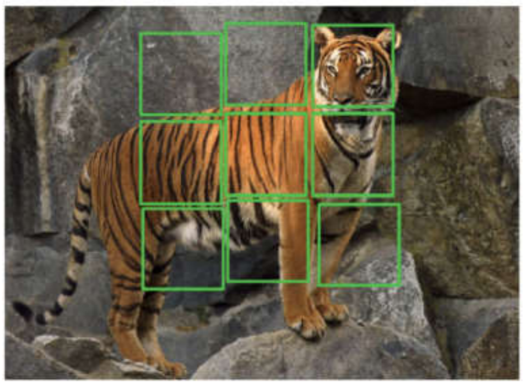


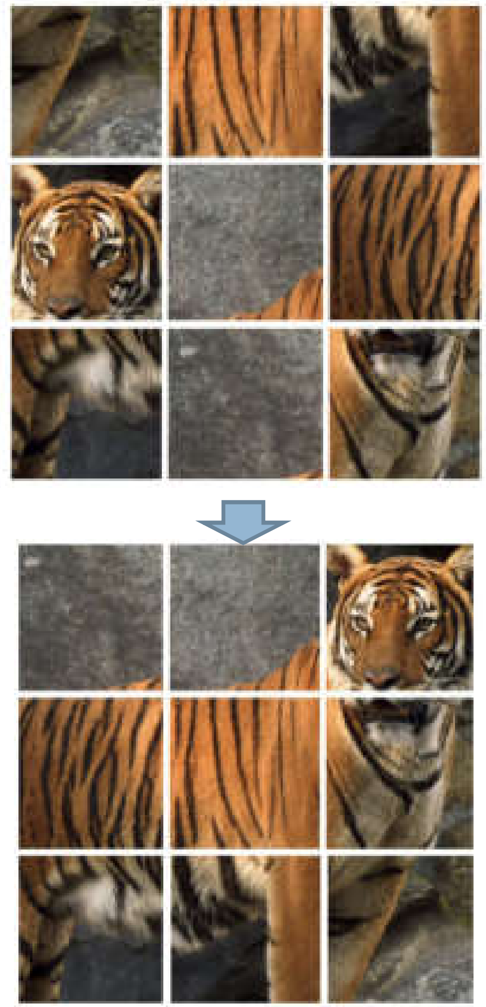




Dorsch (2015) Unsupervised Visual Representation Learning by Context Prediction

Image Jigsaw Puzzle



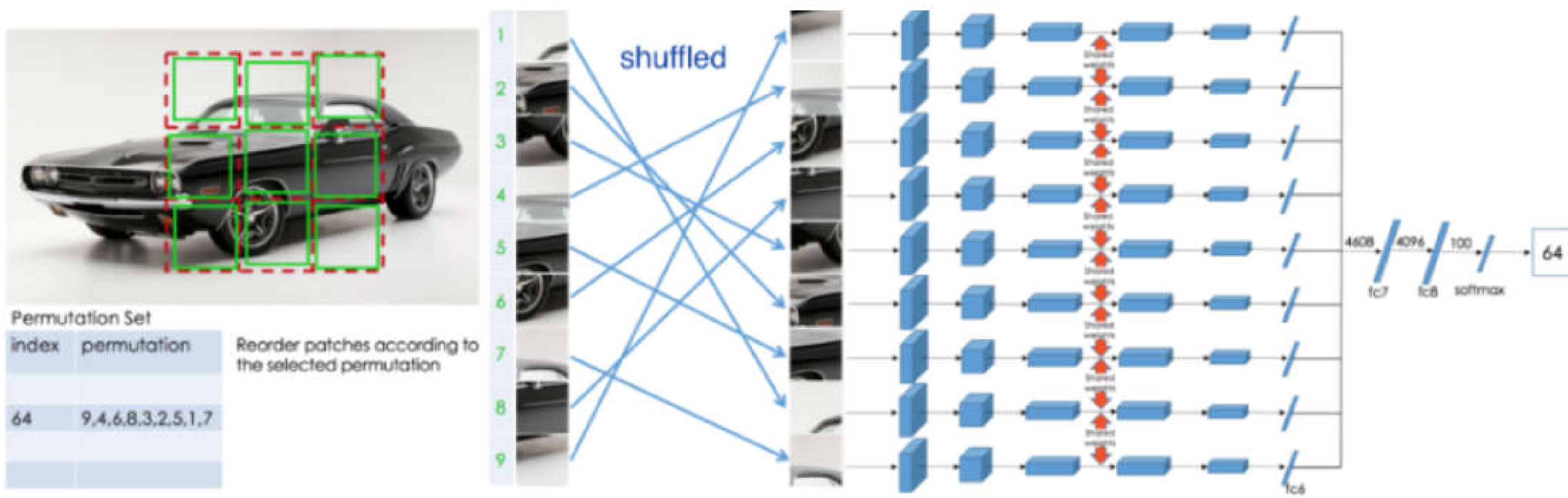


Noroozi (2016) Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles

Image Jigsaw Puzzle

Pretraining data: 9 patches extracted in images

Pretraining task: predict the positions of all 9 patches



Noroozi (2016) Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles

Context Encoders

Pretraining data: remove a random region in images

Pretraining task: fill in a missing piece in the image

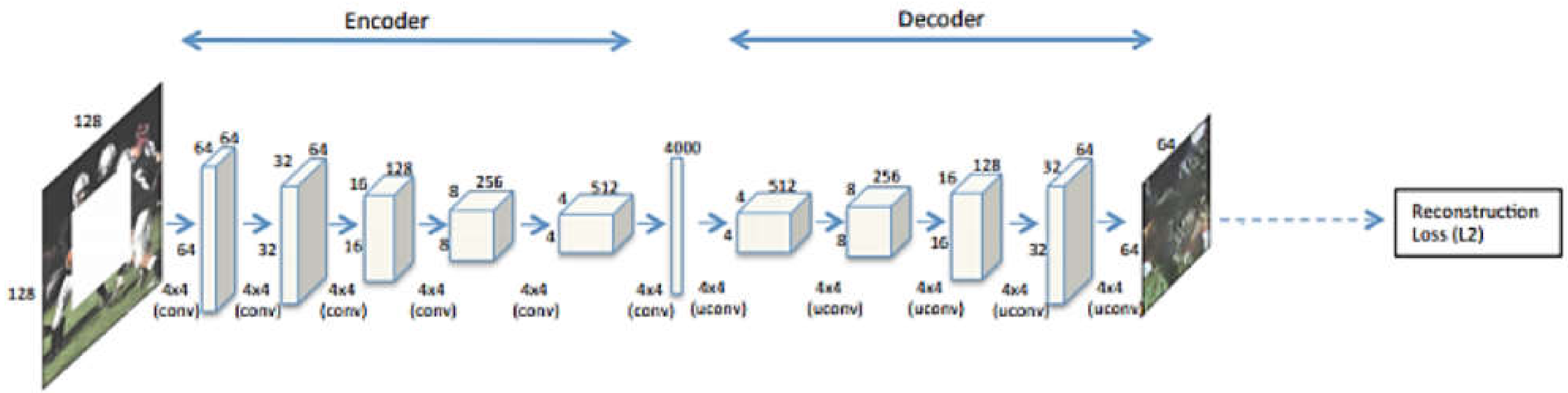
Pathak (2016) Context Encoders: Feature Learning by Inpainting

Context Encoders

an encoder-decoder architecture

A Euclidean ℓ_2 distance is used as the reconstruction loss function L_{rec}

In the downstream task, use the encoder networks as the representation

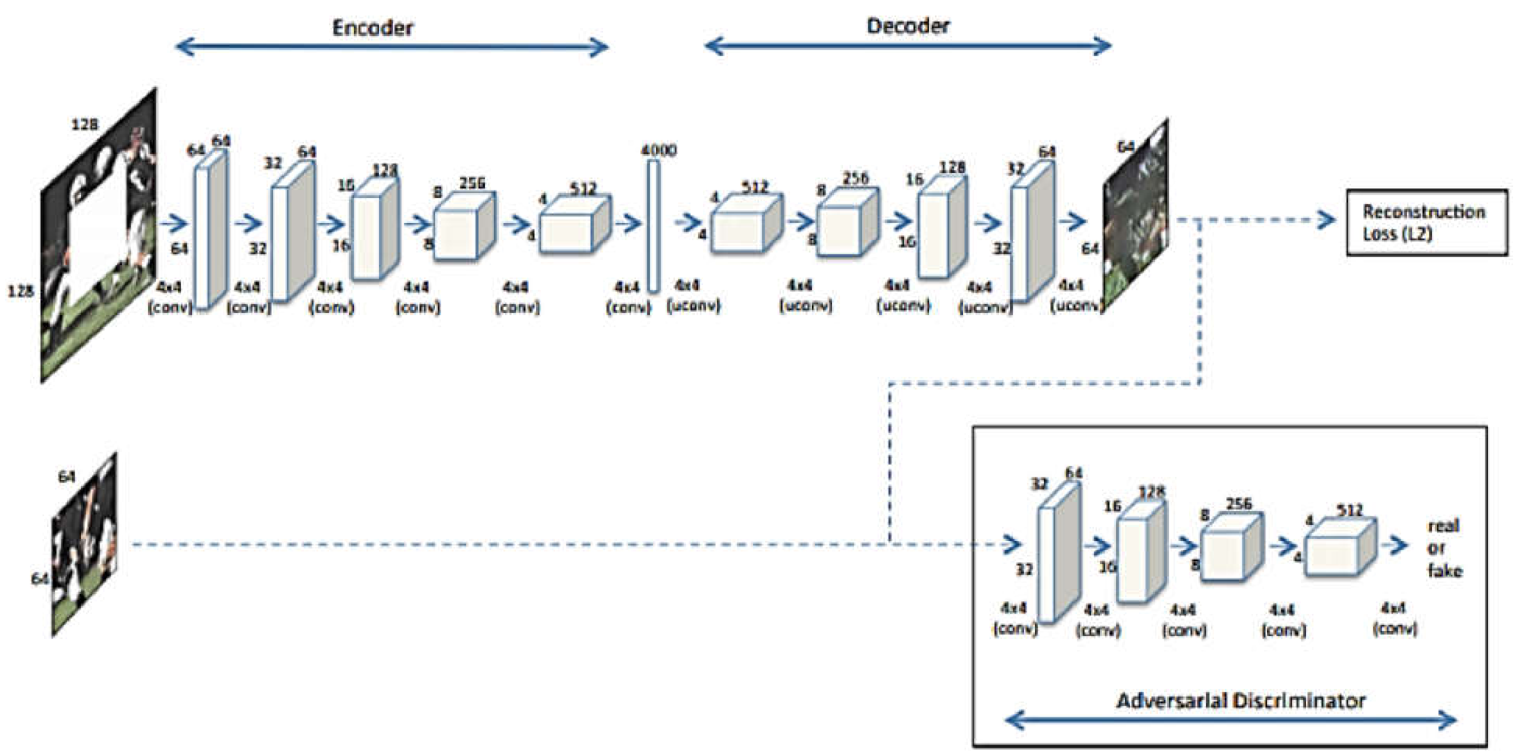


Pathak (2016) Context Encoders: Feature Learning by Inpainting

Context Encoders

Improvement was achieved by adding a GAN branch

A weighted combination of the two losses, i.e., $\lambda_{rec} L_{rec} + \lambda_{gan} L_{gan}$



Context Encoders

Input image

Encoder-decoder with reconstruction loss $\mathcal{L}_{\text{rec}}$

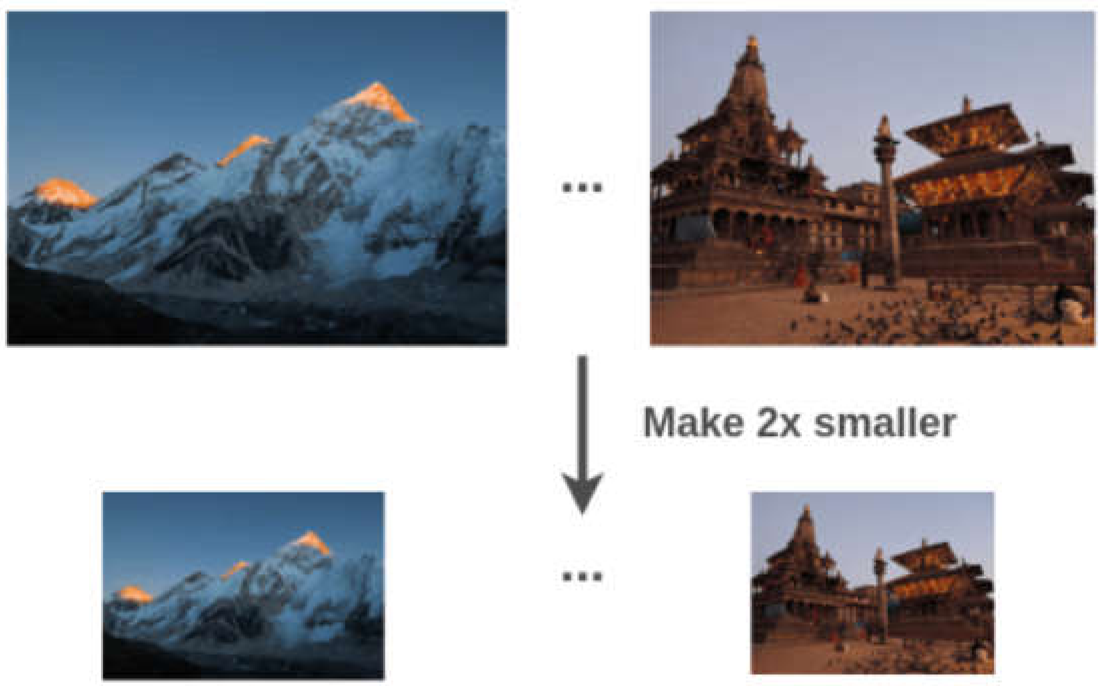
GAN with loss $\mathcal{L}_{\text{gan}}$

Joint loss $\mathcal{L} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{gan}}\mathcal{L}_{\text{gan}}$

Image Super-Resolution

Pretraining data: pairs of regular and downsampled low-resolution images

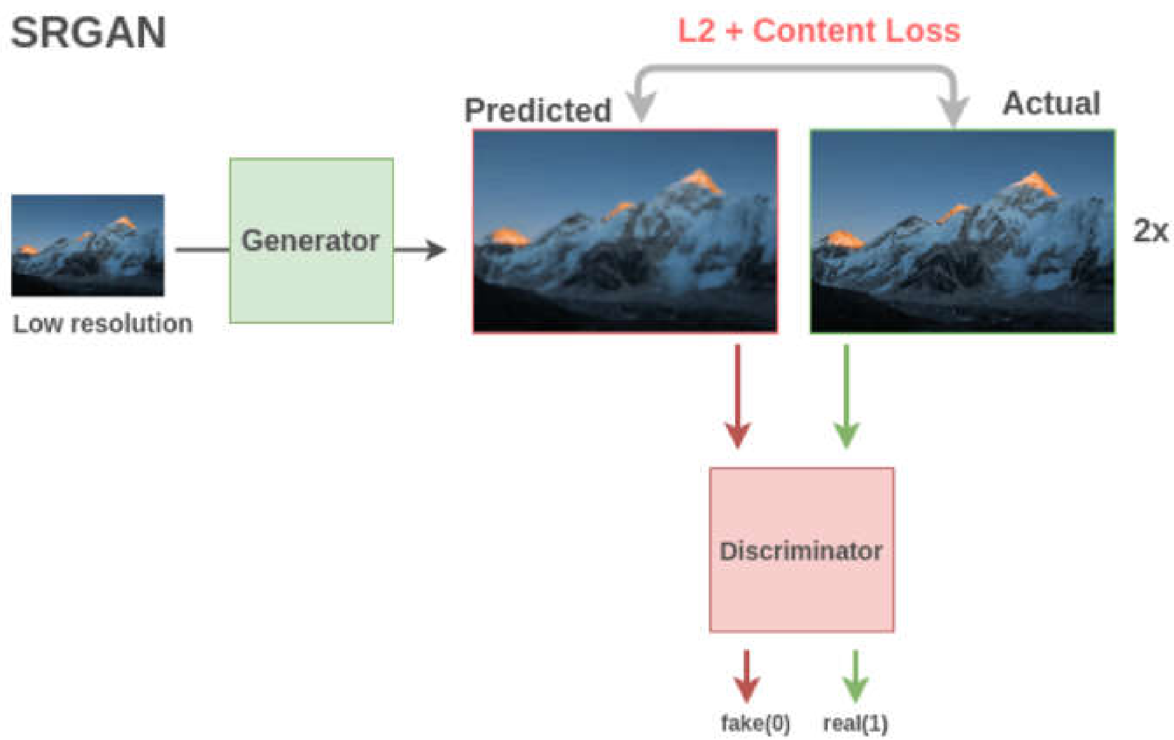
Pretraining task: predict a high-resolution image that corresponds to a downsampled low-resolution image



Ledig (2017) Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

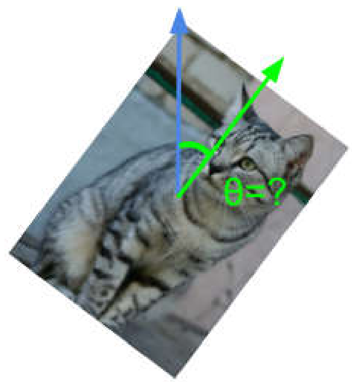
Image Super-Resolution

- A GAN architecture
- The paper did not consider downstream tasks other than super-resolution



Ledig (2017) Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network

image completion



rotation prediction

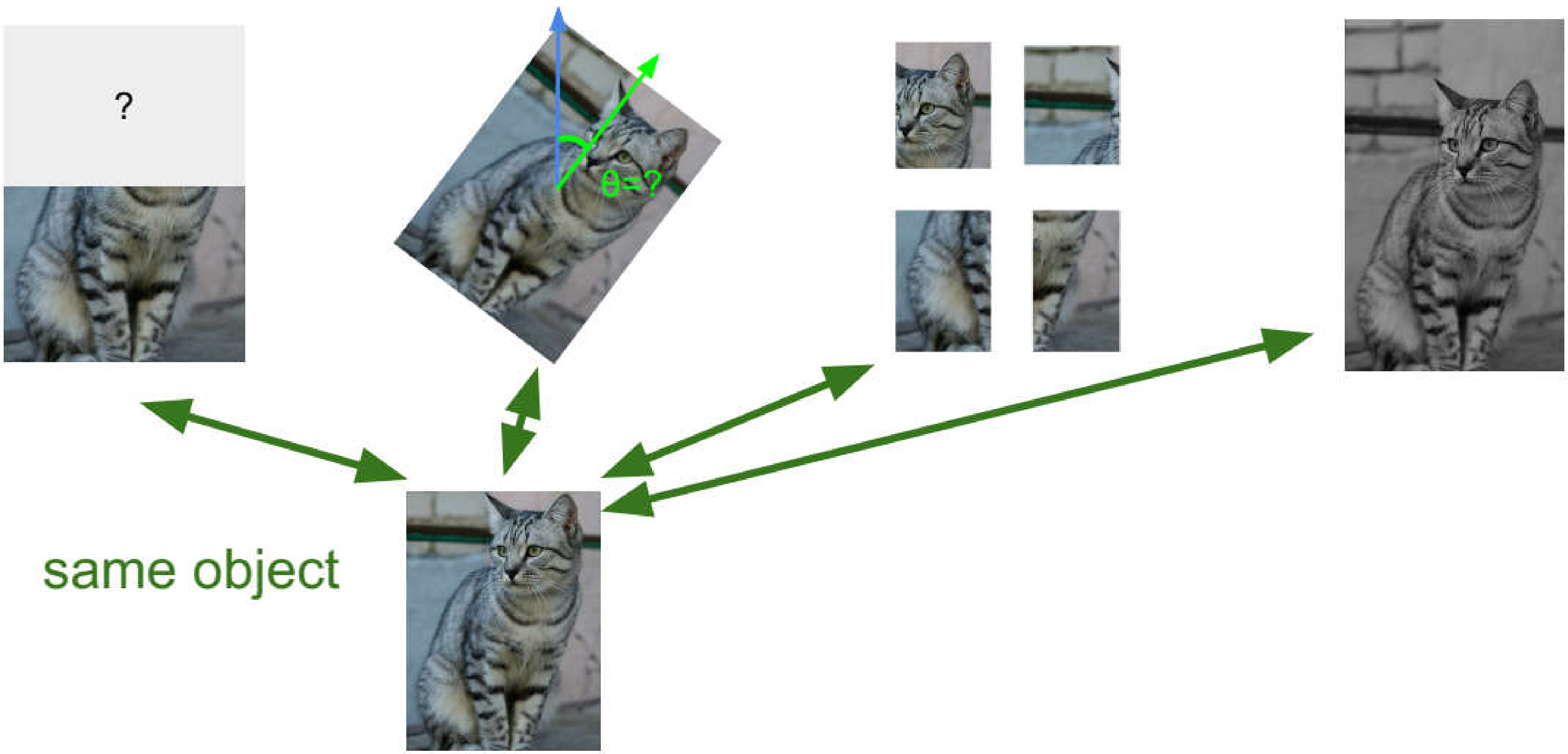
"jigsaw puzzle"

colorization

Learned representations may be tied to a specific pretext task!
Can we come up with a more general pretext task?

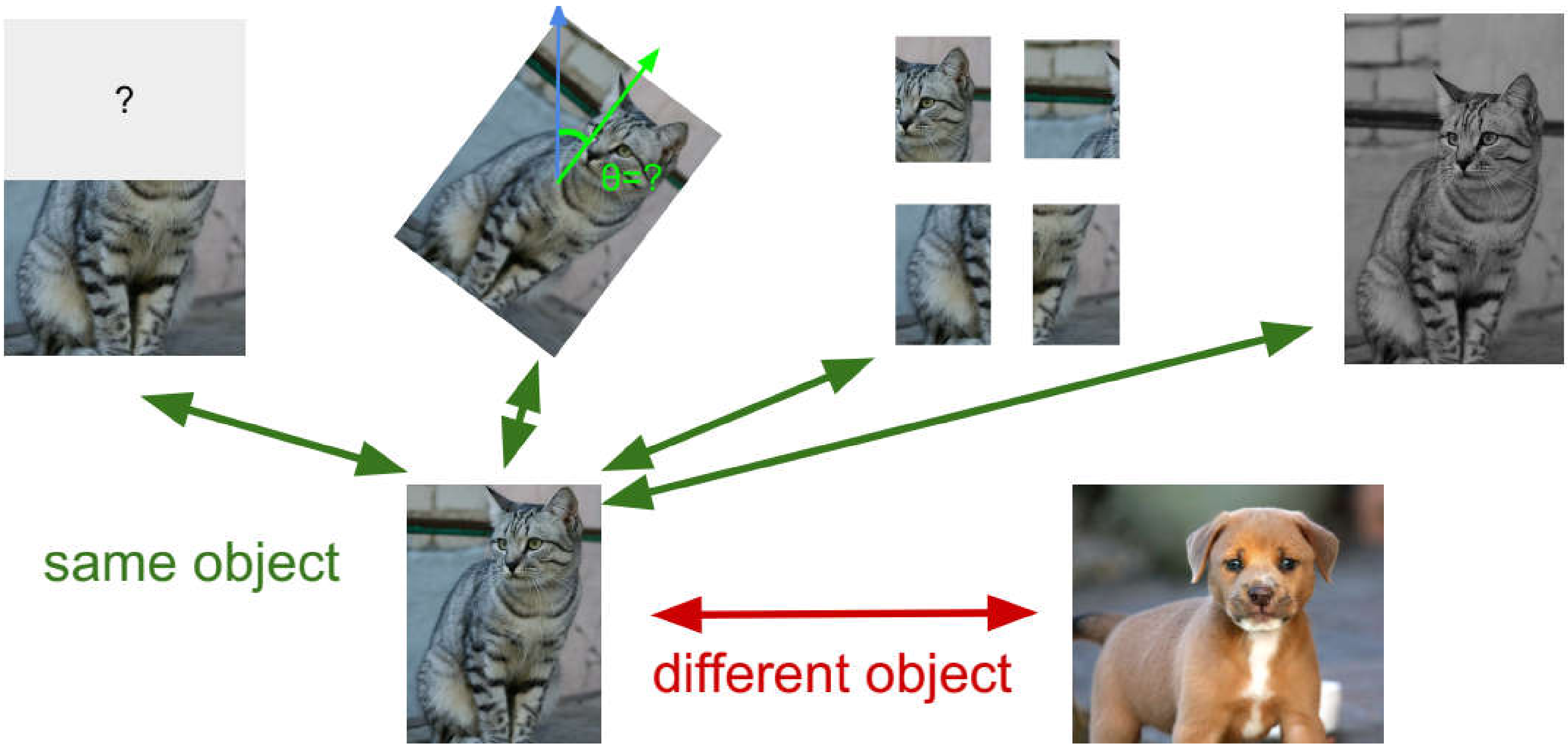
CONTRASTIVE REPRESENTATION LEARNING

Contrastive Representation Learning



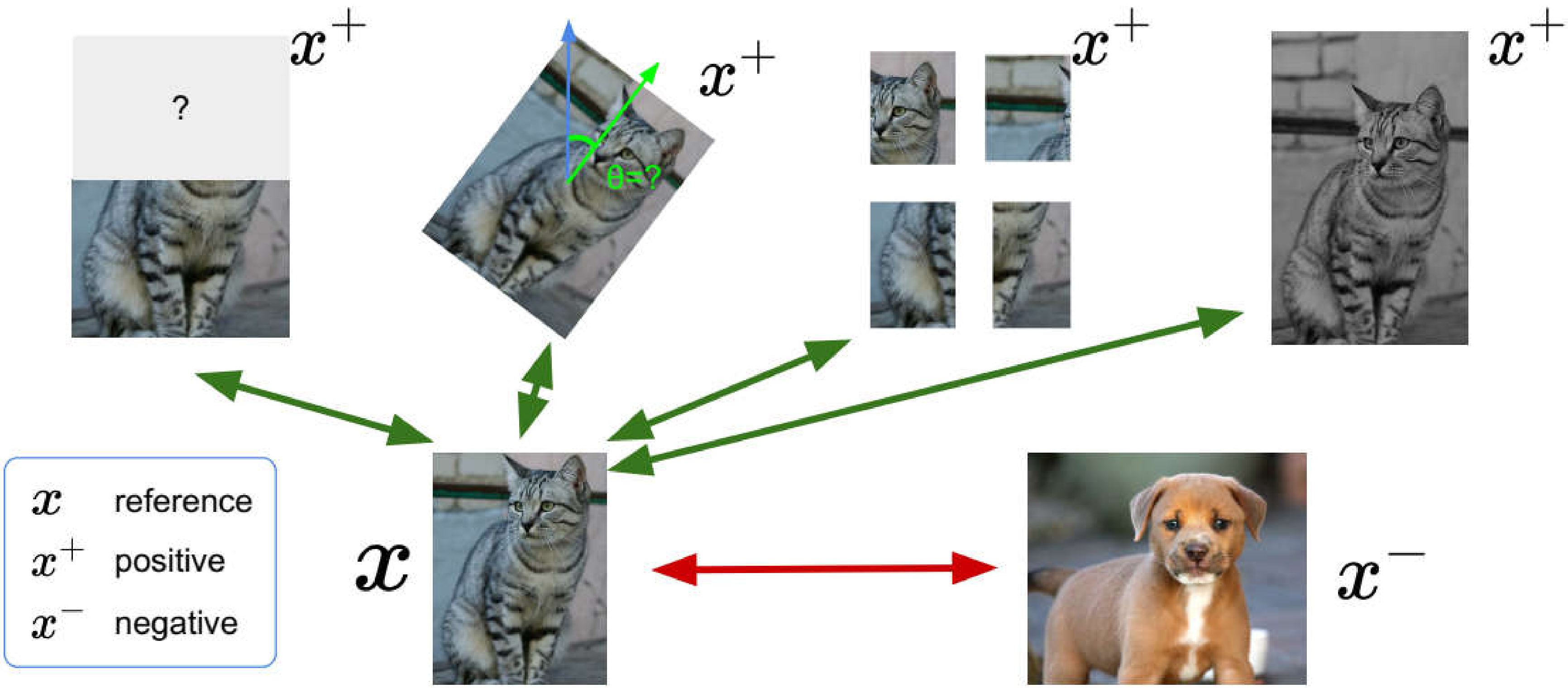
Lectures of deep learning for computer vision course (CS231n course -Stanford university)

Contrastive Representation Learning



Lectures of deep learning for computer vision course (CS231n course -Stanford university)

Contrastive Representation Learning



Lectures of deep learning for computer vision course (CS231n course -Stanford university)

Contrastive Representation Learning formulation

Encoder function

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

Contrastive Representation Learning formulation

$$\text{score}(f(x), f(x^+)) >> \text{score}(f(x), f(x^-))$$

Loss function given 1 positive sample and N - 1 negative samples:

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+))}{\exp(s(f(x), f(x^+)) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))} \right]$$

Lectures of deep learning for computer vision course (CS231n course -Stanford university)

Contrastive Representation Learning formulation

$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+)))}{\exp(s(f(x), f(x^+))) + \sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))} \right]$$

Lectures of deep learning for computer vision course (CS231n course -Stanford university)

Contrastive Representation Learning

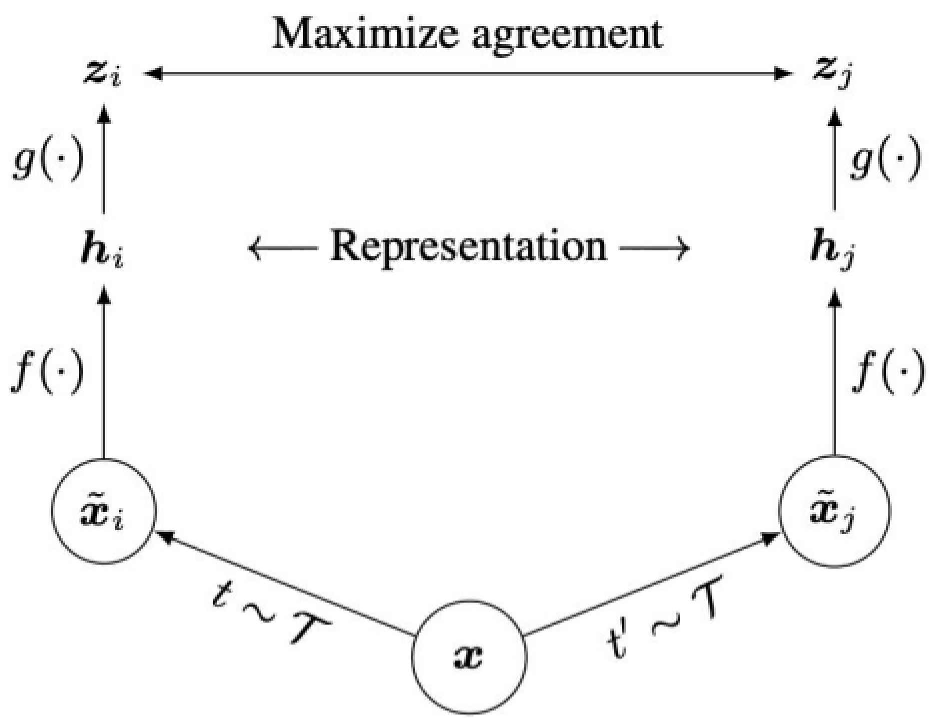
$$L = -\mathbb{E}_X \left[\log \frac{\exp(s(f(x), f(x^+)))}{\underbrace{\exp(s(f(x), f(x^+)))}_{\text{score for the positive pair}} + \underbrace{\sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-)))}_{\text{score for the N-1 negative pairs}}} \right]$$

Contrastive Representation Learning

$$L = -\mathbb{E}_X \left[\log \frac{\overline{\exp(s(f(x), f(x^+))}}{\underbrace{\exp(s(f(x), f(x^+))}_{\text{score for the positive pair}} + \underbrace{\sum_{j=1}^{N-1} \exp(s(f(x), f(x_j^-))}_{\text{score for the N-1 negative pairs}}} \right]$$

Cross entropy loss for a N-way softmax classifier!
I.e., learn to find the positive sample from the N samples

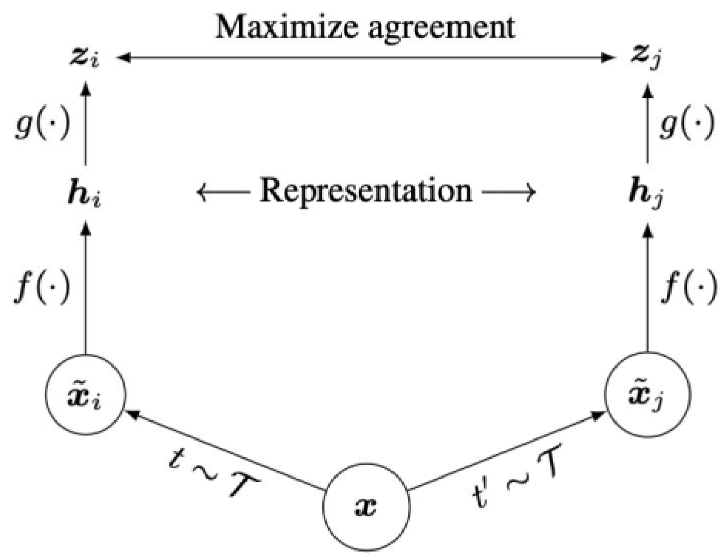
SimCLR: A Simple Framework for Contrastive Learning



Ting Chen et al , "A Simple Framework for Contrastive Learning of Visual Representations", 2020

SimCLR: A Simple Framework for Contrastive Learning

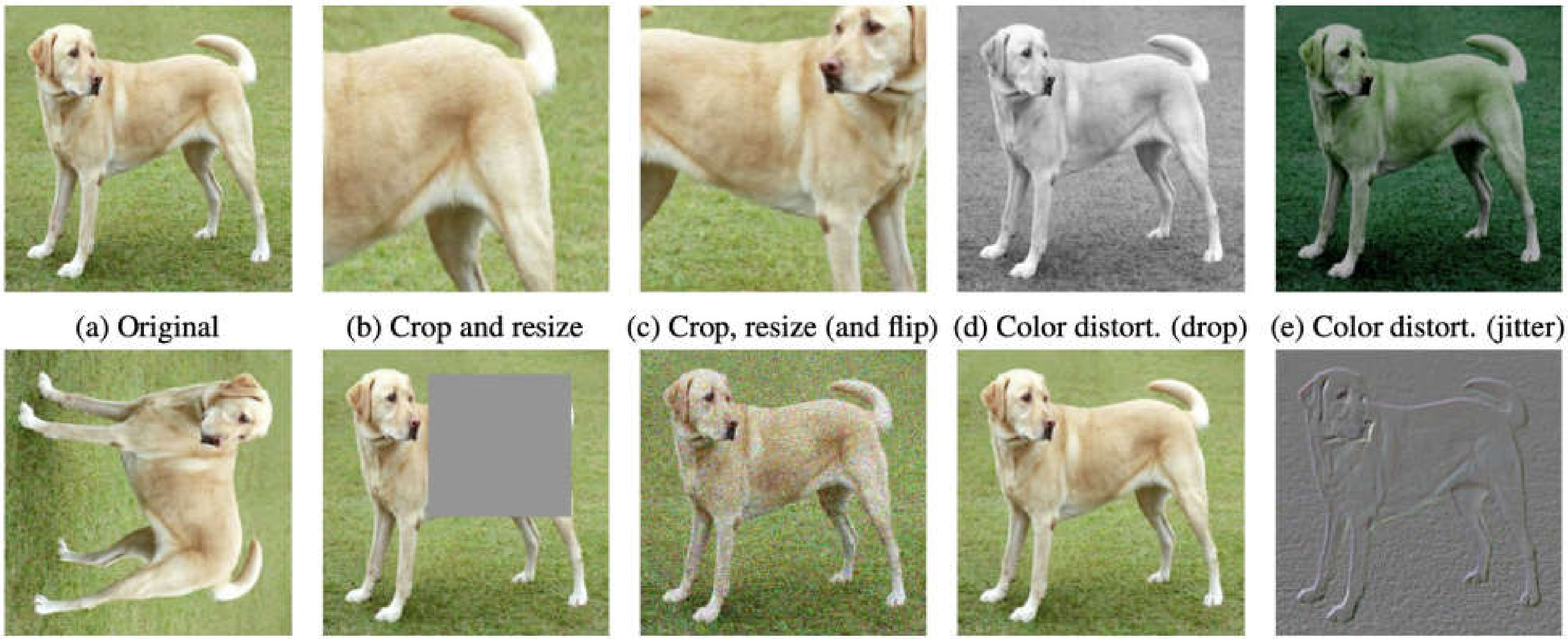
Use a projection network ***g(·)*** to project features to a space where contrastive learning is applied



$$s(u,v) = \frac{u^T v}{||u||\,||v||}$$

Ting Chen et al , “A Simple Framework for Contrastive Learning of Visual Representations”, 2020

SimCLR



(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

Ting Chen et al , “A Simple Framework for Contrastive Learning of Visual Representations”, 2020

SimCLR

SimCLR

Generate a positive pair by sampling data augmentation functions

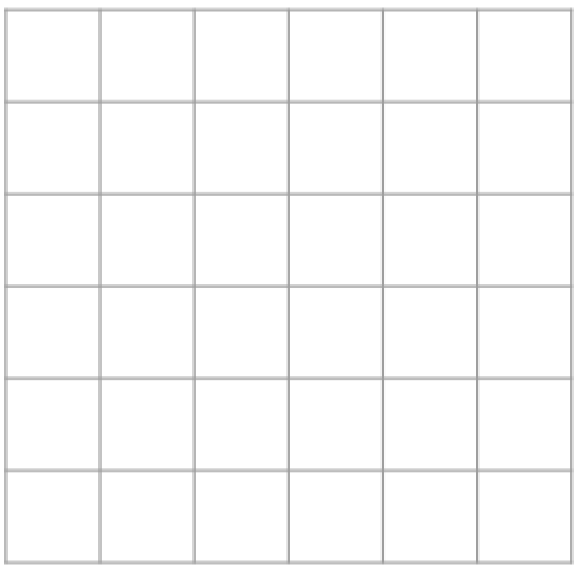
Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, constant τ, structure of f, g, $\mathcal{T}$.
for sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    $\boxed{\text{draw two augmentation functions } t \sim \mathcal{T},\, t' \sim \mathcal{T}}$
    # the first augmentation
    $\boxed{\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)}$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\boxed{\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)}$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^{\top} \boldsymbol{z}_j / (\|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i,j)$ **as** $\ell(i,j) = -\log \frac{\exp(s_{i,j}/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k}/\tau)}$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network $f(\cdot)$, and throw away $g(\cdot)$

SimCLR

$$s_{i,j} = \frac{z_i^T z_j}{||z_i|| \, ||z_j||}$$

"Affinity matrix"

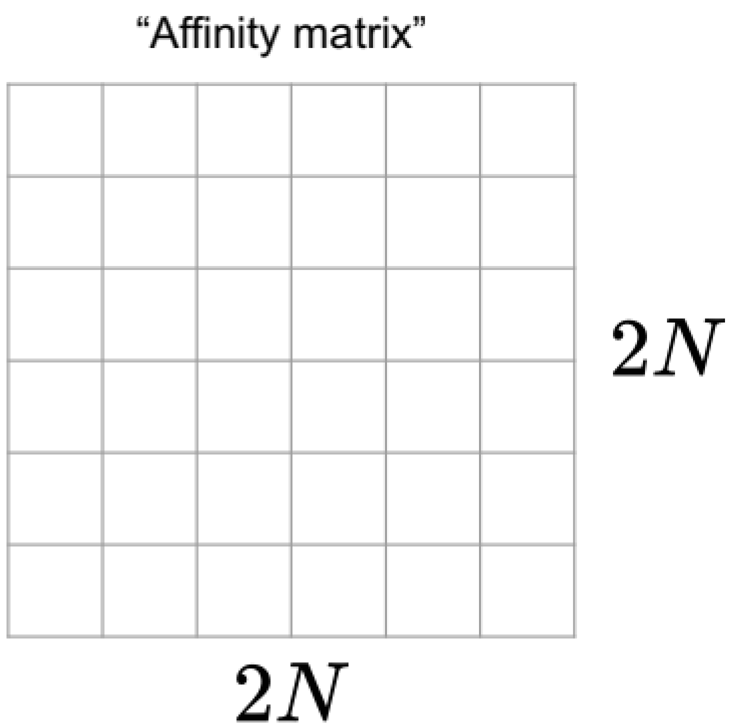


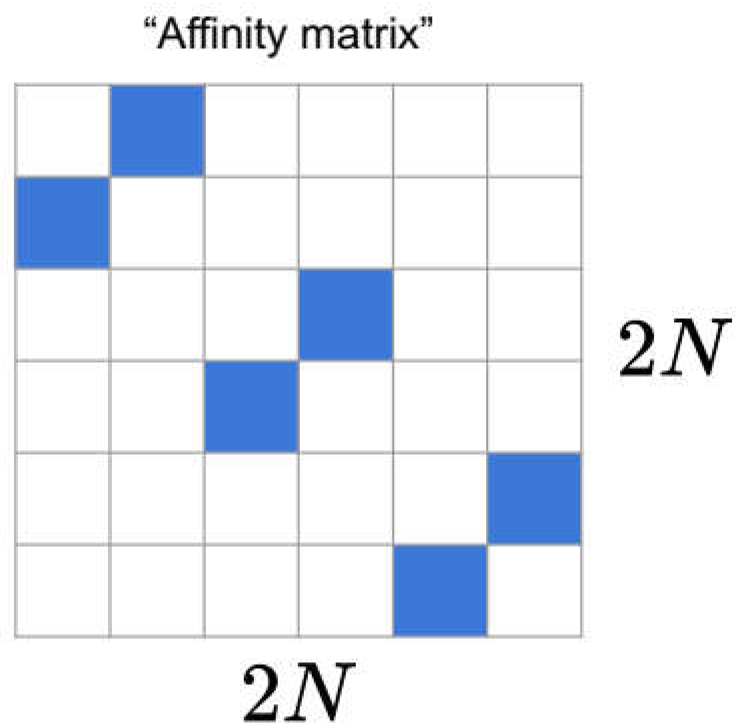
$2N$

$2N$

SimCLR

$$s_{i,j} = \frac{z_i^T z_j}{||z_i|| \, ||z_j||}$$





Lectures of deep learning for computer vision course (CS231n course -Stanford university)