CS480/680: Introduction to Machine Learning

Lecture 17: Self-Supervised Learning

Hongyang Zhang

UNIVERSITY OF WATERLOO

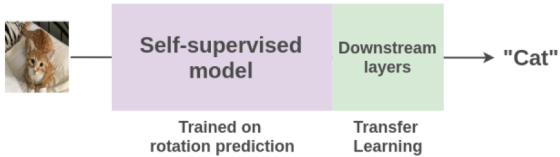
July 30, 2025

Learning Paradigms

- Supervised learning – learning with labeled data
  - Collect a small dataset with labels (labels are expensive)
  - Examples: SVM, Logistic Regression, LLM finetuning, etc.
- Unsupervised learning – learning with unlabeled data
  - Collect a large dataset without label (unlabeled data are cheap)
  - Examples: LLM pretraining, GAN, etc.
- Self-supervised learning – a subclass of unsupervised learning
  - Goal: Learn useful representations through pretraining tasks for downstream tasks
  - Example: LLM pretraining (predicting masked tokens)

Self-Supervised Learning

- Self-supervised learning steps
  - Pretraining/Pretext step: build a task where the label is pseudo and is constructed from the unlabeled data (e.g., predict the rotation degree of rotated images)
  - Downstream step:
    - Fine-tuning protocol: all trainable parameters
    - Linear evaluation protocol: Fix the representation and fine-tuning topping layers



Why self-supervised learning?

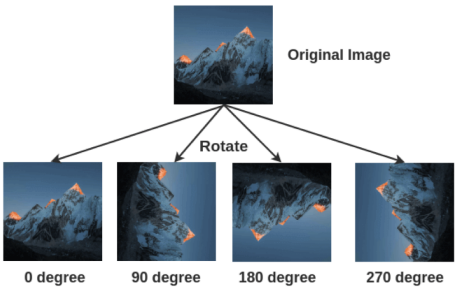
- Why self-supervised learning?
  - Creating labeled datasets for each task is an expensive
  - Vast amount of unlabeled data on the internet (images, videos, text)
  - Self-supervised learning will not overfit
- Challenges for self-supervised learning
  - How to select a suitable pretraining task for an application
  - There is no golden rule for comparison of learned feature representations

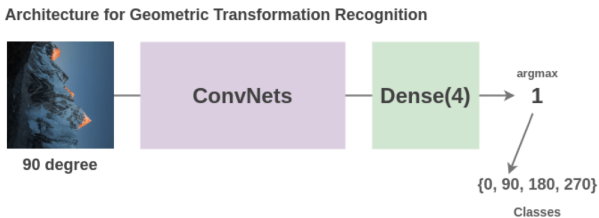
Outline (Pretraining Tasks)

- Geometric transformation recognition
  - Image rotation
- Patches
  - Relative patch position
  - Image jigsaw puzzle
- Generative modeling
  - Context encoders
  - Image colorization
  - Cross-channel prediction
  - Image super-resolution
- Contrastive learning
  - SimCLR

Image Rotation

- Pretraining data: images rotated by a multiple of 90 degree at random
  - This corresponds to four rotated images at 0, 90, 180, and 270 degrees
- Pretraining task: train a model to predict the rotation degree that was applied





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

Image Rotation

- A single ConvNet model is used to predict one of the four rotations
- The model needs to understand the location and type of the objects in images to determine the rotation degree

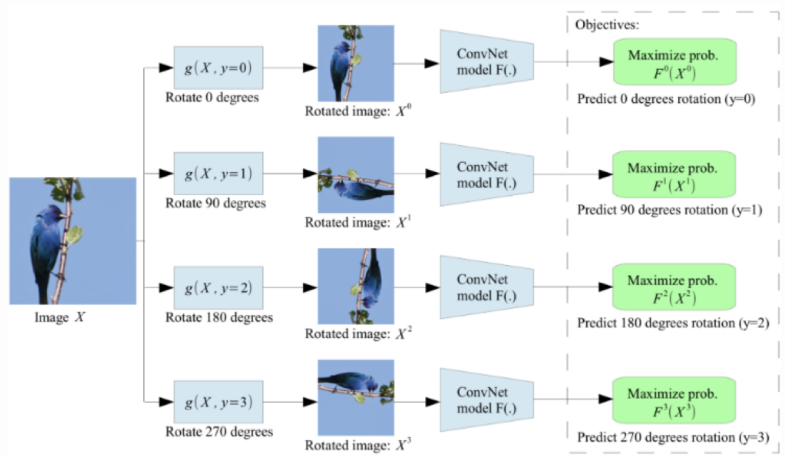


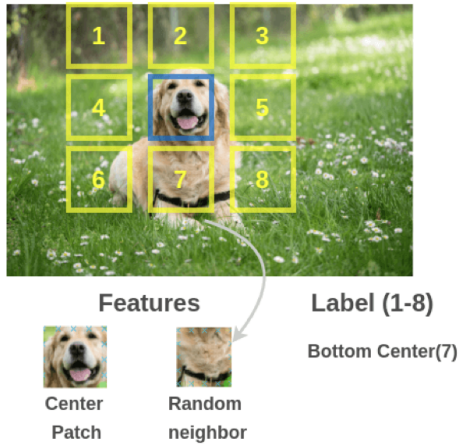
Image Rotation Evaluation

- Evaluation on the PASCAL VOC dataset for classification, detection, and segmentation tasks
  - The model (RotNet) is trained in SSL manner, and fine-tuned afterwards
  - The learned features are not as good as the supervised learned features based on transfer learning from ImageNet, but they demonstrate a potential

	Classification (%mAP)		Detection (%mAP)	Segmentation (%mIoU)
Trained layers	fc6-8	all	all	all
Supervised feature learning: ImageNet labels	78.9	79.9	56.8	48.0
Random		53.3	43.4	19.8
Random rescaled Krähenbühl et al. (2015)	39.2	56.6	45.6	32.6
Egomotion (Agrawal et al., 2015)	31.0	54.2	43.9	
Context Encoders (Pathak et al., 2016b)	34.6	56.5	44.5	29.7
Tracking (Wang & Gupta, 2015)	55.6	63.1	47.4	
Context (Doersch et al., 2015)	55.1	65.3	51.1	
Colorization (Zhang et al., 2016a)	61.5	65.6	46.9	35.6
BIGAN (Donahue et al., 2016)	52.3	60.1	46.9	34.9
Jigsaw Puzzles (Noroozi & Favaro, 2016)	-	67.6	53.2	37.6
NAT (Bojanowski & Joulin, 2017)	56.7	65.3	49.4	
Split-Brain (Zhang et al., 2016b)	63.0	67.1	46.7	36.0
ColorProxy (Larsson et al., 2017)		65.9		38.4
Counting (Noroozi et al., 2017)	-	67.7	51.4	36.6
Proposed self-supervised feature learning: (Ours) RotNet	**70.87**	**72.97**	**54.4**	**39.1**

Relative Patch Position

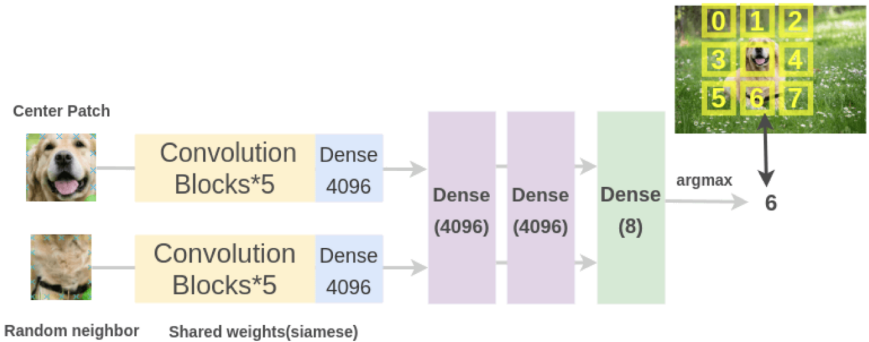
- Pretraining data: multiple patches extracted from images
- Pretraining task: train a model to predict the relationship between the patches



Dorsch (2015) Unsupervised Visual Representation Learning by Context Prediction

Relative Patch Position

- The patches are inputted into two ConvNets with shared weights
- The model needs to understand the spatial context of images, in order to predict the relative positions between the patches



Relative Patch Position

- The training patches are sampled in the following way:
  - Randomly sample the first patch, and consider it the middle of a 3x3 grid
  - Sample from 8 neighboring locations of the first central patch (blue patch)
- To avoid the model only catching low-level trivial information:
  - Add gaps between the patches
  - Add small jitters to the positions of the patches
  - Randomly downsample some patches to reduced resolution, and then upsample
  - Randomly drop 1 or 2 color channels for some patches

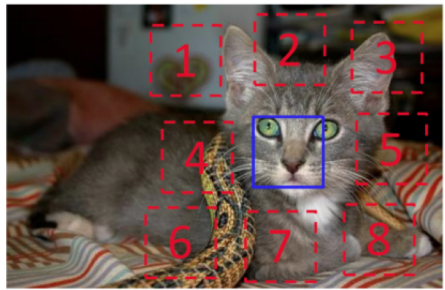
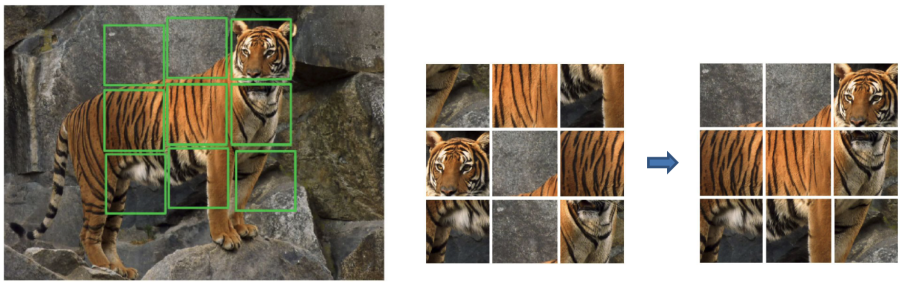


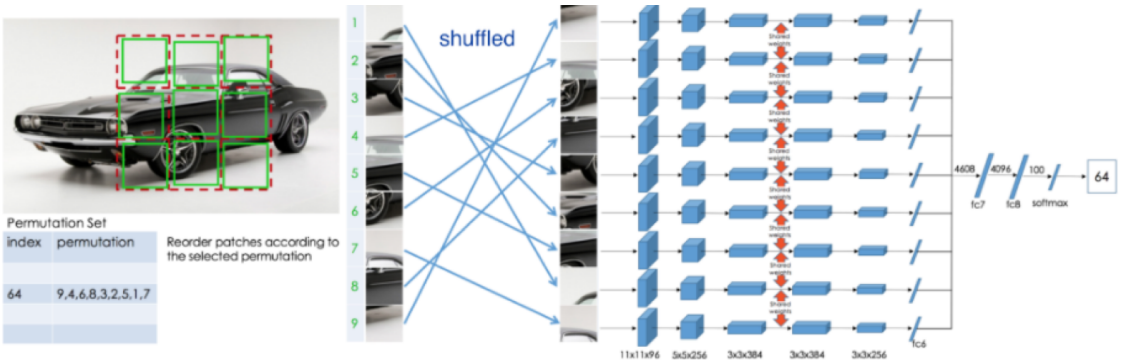
Image Jigsaw Puzzle

- Pretraining data: 9 patches extracted in images (similar to the previous approach)
- Pretraining task: predict the positions of all 9 patches
  - This approach uses the grid of 3-by-3 patches and solves a jigsaw puzzle



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

Image Jigsaw Puzzle



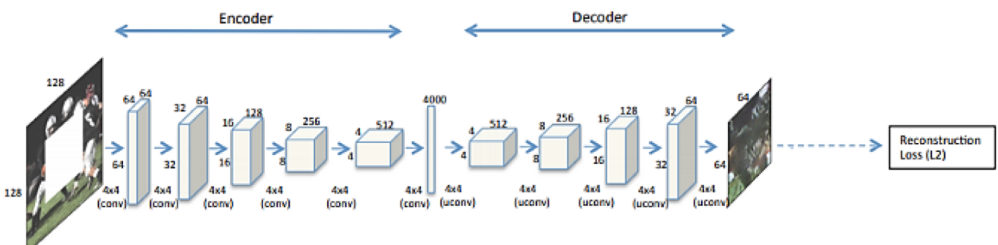
Context Encoders

- 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

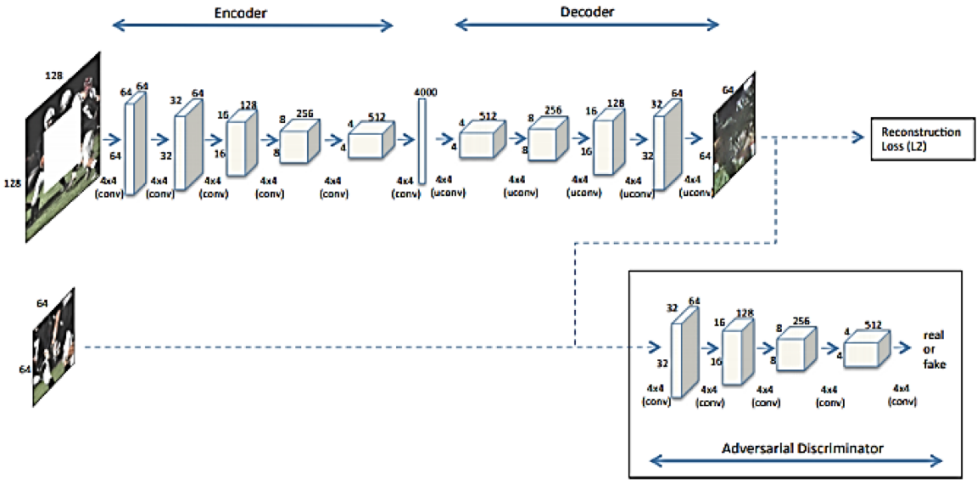
Context Encoders

- The initially considered model uses 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



Context Encoders

- Improvement was achieved by adding a GAN branch
- A weighted combination of the two losses, i.e., $\lambda_{rec} L_{rec} + \lambda_{gan} L_{gan}$

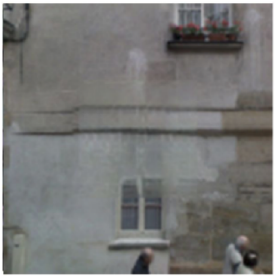


Context Encoders

Input image

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

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



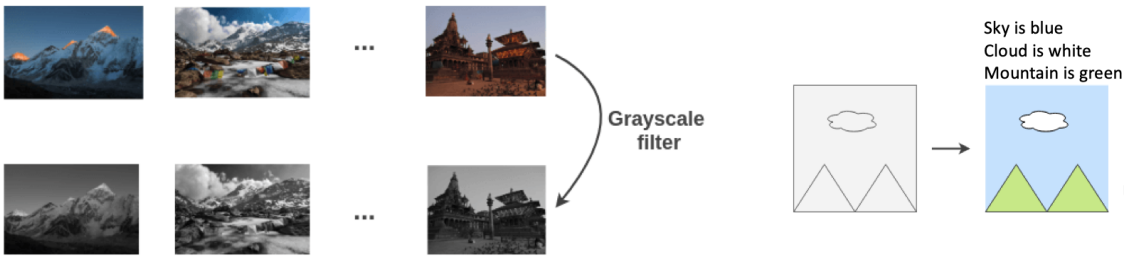
Joint loss $\mathcal{L} = \lambda_{\text{rec}}\mathcal{L}_{\text{rec}} + \lambda_{\text{gan}}\mathcal{L}_{\text{gan}}$

Context Encoders

Pretraining Method	Supervision	Pretraining time	Classification	Detection	Segmentation
ImageNet [26]	1000 class labels	3 days	**78.2%**	**56.8%**	**48.0%**
Random Gaussian	initialization	$<$ 1 minute	53.3%	43.4%	19.8%
Autoencoder	-	14 hours	53.8%	41.9%	25.2%
Agrawal *et al.* [1]	egomotion	10 hours	52.9%	41.8%	-
Doersch *et al.* [7]	context	4 weeks	55.3%	**46.6%**	-
Wang *et al.* [39]	motion	1 week	**58.4%**	44.0%	-
Ours	context	14 hours	56.5%	44.5%	**29.7%**

Image Colorization

- Pretraining data: pairs of color and grayscale images
- Pretraining task: predict the colors of the objects in grayscale images



Zhang (2016) Colorful Image Colorization

Image Colorization

- An encoder-decoder architecture with convolutional layers
- ℓ_2 loss between the actual color image and the predicted colorized image
- In the downstream task, use the encoder as the representation

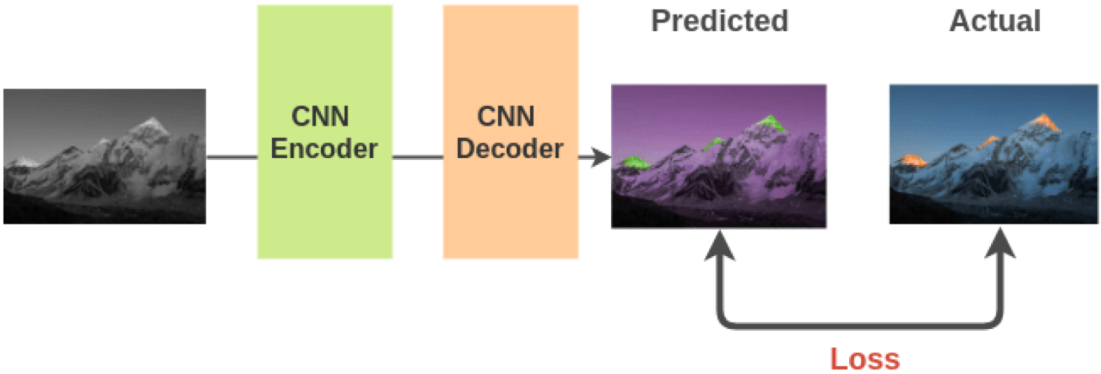
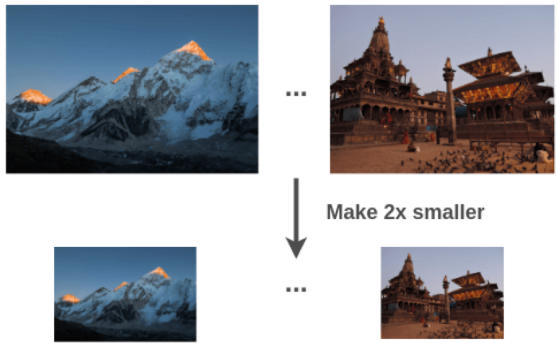


Image Super-Resolution

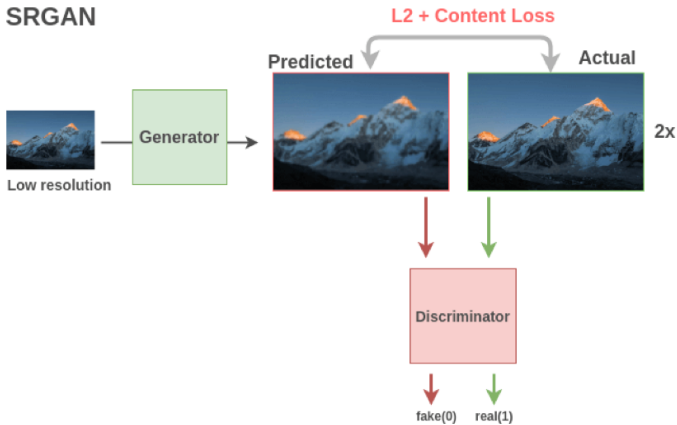
- Pretraining data: pairs of regular and downsampled low-resolution images
- Pretraining task: predict a high-resolution image that corresponds to a downsampled low-resolution image



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

Contrastive Learning

SimCLR

A Simple Framework for Contrastive Learning of Visual Representations

Ting Chen [1] Simon Kornblith [1] Mohammad Norouzi [1] Geoffrey Hinton [1]

Abstract

This paper presents *SimCLR*: a simple framework for contrastive learning of visual representations. We simplify recently proposed contrastive self-supervised learning algorithms without requiring specialized architectures or a memory bank. In order to understand what enables the contrastive prediction tasks to learn useful representations, we systematically study the major components of our framework. We show that (1) composition of data augmentations plays a critical role in defining effective predictive tasks, (2) introducing a learnable nonlinear transformation between the representation and the contrastive loss substantially improves the quality of the learned representations, and (3) contrastive learning benefits from larger batch sizes and more training steps compared to supervised learning. By combining these findings, we are able to considerably outperform previous methods for self-supervised and semi-supervised learning on ImageNet. A linear classifier trained on self-supervised representations learned by SimCLR achieves 76.5% top-1 accuracy, which is a 7% relative improvement over previous state-of-the-art, matching the performance of a supervised ResNet-50. When fine-tuned on only 1% of the labels, we achieve 85.8% top-5 accuracy, outperforming AlexNet with 100× fewer labels. [1]

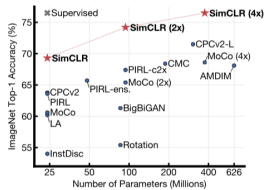
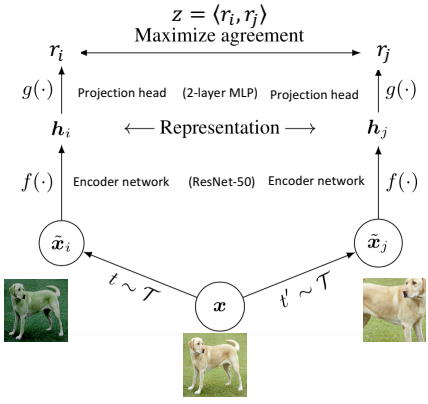


Figure 1. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. Our method, SimCLR, is shown in bold.

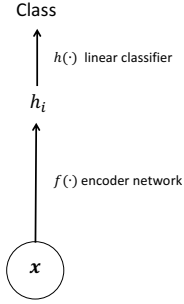
However, pixel-level generation is computationally expensive and may not be necessary for representation learning. Discriminative approaches learn representations using objective functions similar to those used for supervised learning, but train networks to perform pretext tasks where both the inputs and labels are derived from an unlabeled dataset. Many such approaches have relied on heuristics to design pretext tasks (Doersch et al., 2015; Zhang et al., 2016; Noroozi & Favaro, 2016; Gidaris et al., 2018), which could limit the

Chen (2020) A Simple Framework for Contrastive Learning of Visual Representations

Structure of SimCLR



a) Training

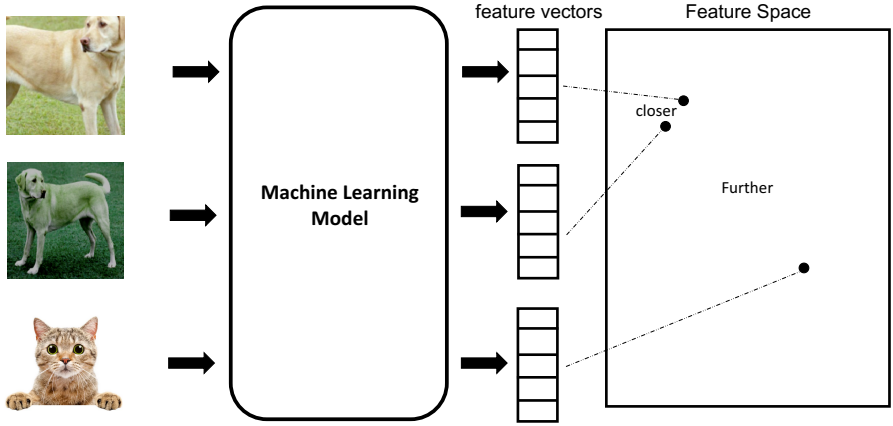


b) Testing

- But how to measure agreement? By comparison!

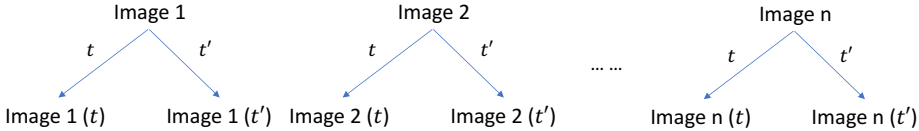
Chen (2020) A Simple Framework for Contrastive Learning of Visual Representations

How to measure agreement

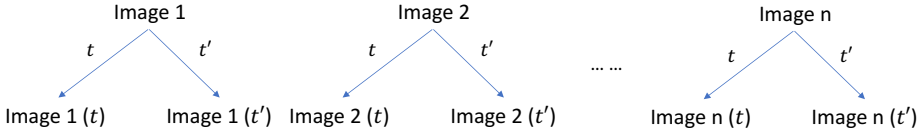


- Distance of images of same content $\leq$ Distance of images of different contents
- Similarity of images of same content $\geq$ Similarity of images of different contents

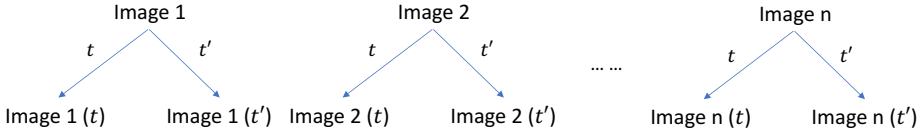
Loss Function



Loss Function



Loss Function



Performance

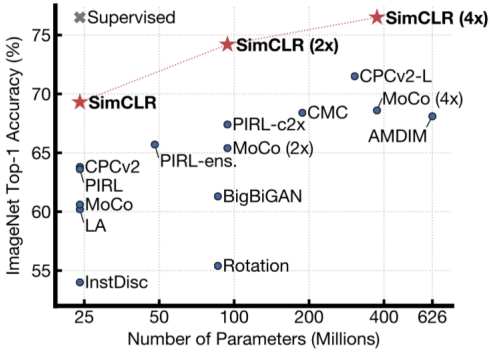


Figure 1. ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. Our method, SimCLR, is shown in bold.

- The 1st method that is comparable with supervised learning on ImageNet by linear evaluation protocol

Performance

- Experimental results on 10 image datasets
- SimCLR outperformed supervised models on most datasets

	Food	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech-101	Flowers
Linear evaluation:												
SimCLR (ours)	**76.9**	**95.3**	80.2	48.4	**65.9**	60.0	61.2	**84.2**	**78.9**	89.2	**93.9**	**95.0**
Supervised	75.2	**95.7**	**81.2**	**56.4**	64.9	**68.8**	**63.8**	83.8	**78.7**	**92.3**	**94.1**	94.2
Fine-tuned:												
SimCLR (ours)	**89.4**	**98.6**	**89.0**	**78.2**	**68.1**	**92.1**	**87.0**	**86.6**	**77.8**	92.1	**94.1**	97.6
Supervised	88.7	98.3	**88.7**	**77.8**	67.0	91.4	**88.0**	86.5	**78.8**	**93.2**	**94.2**	**98.0**
Random init	88.3	96.0	81.9	**77.0**	53.7	91.3	84.8	69.4	64.1	82.7	72.5	92.5


Questions
?
?
Answers
?