Contrastive Learning

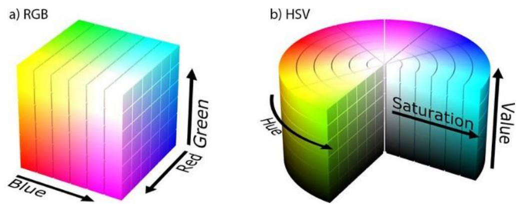
221124 Advanced Computer Vision

Sohee Kim

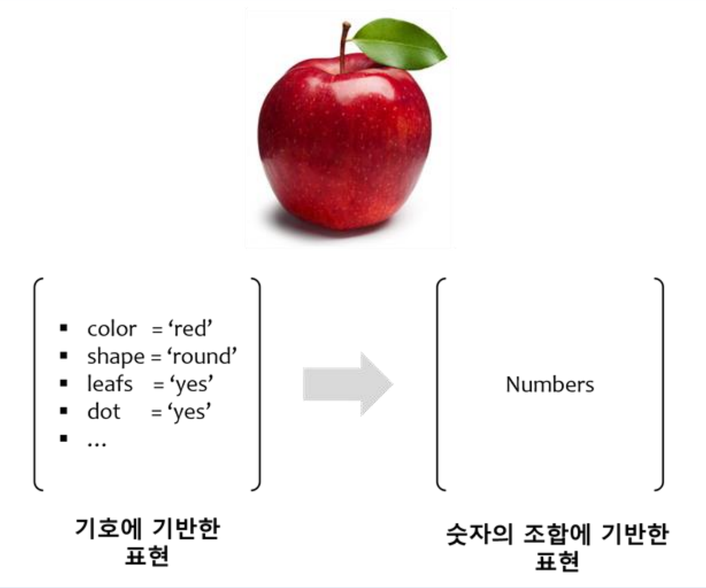
◇ Background

❖ Representation Learning

- A class of machine learning approaches that allow a system to discover the representations required for feature detection or classification from raw data
- *"raw 데이터에서 detection이나 classification에 필요한 표현을 자동으로 검색할 수 있도록 하는 일련의 기술"*
  - **Learning useful reprentation based on input data**
- Representation : Different ways to view data to encode or describe it



➢ Image : RGB format / HSV format



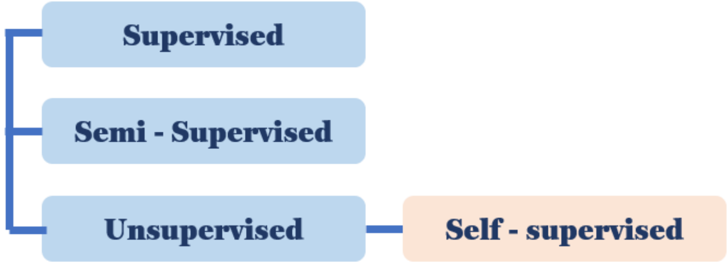
- Two approaches
  - Generative(생성) approaches – generate similar image
    - Pixel level generation : computationally expensive
  - Discriminative(판별) approaches – recognize tag, sort data
    - Train networks to perform pretext tasks : limit the generality of learned representations

https://medium.com/neurosapiens/segmentation-and-classification-with-hsv-8f2406c62b39

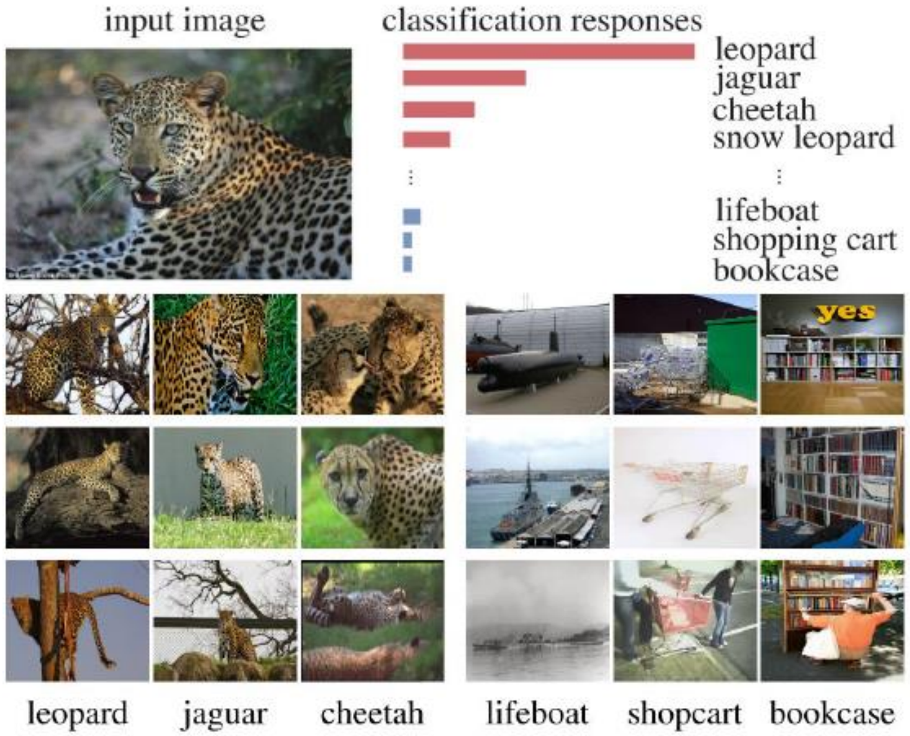
https://steemit.com/representation-learning/@hugmanskj/representation-learning

◇ Background

❖ Self-supervised Learning



- Deep learning – need enough quality and data
    - Labeling process is essential → Difficult to collect sufficiently
- A field of Unsupervised – learns features from unlabeled input data train
- A method designed to obtain its **own label** using information that can be obtained from data



> Result of image classification model based on supervised

❖ Motivation of Contrastive Learning

- Input image : cheetah

  ⇒ Classification response : Classification rates of <u>leopards, jaguars,</u> etc. are higher than those of boats or shopping carts

    - It can be seen that similar features are acting on <u>cheetahs, raopards, jaguars, etc.,</u> and these well-extracted feature values start with the assumption that they will have similarity information between instances

https://daeun-computer-uneasy.tistory.com/37

◇ Background

❖ Contrastive Learning

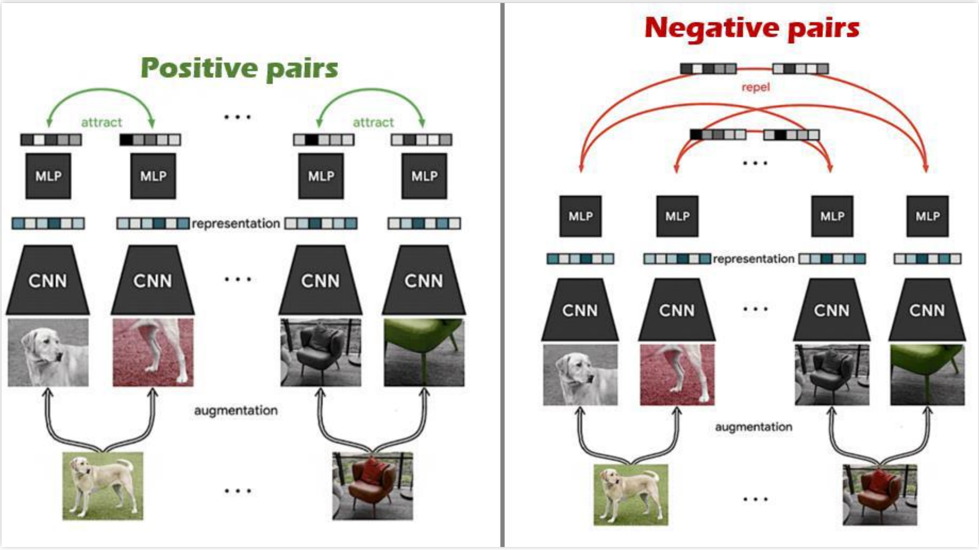
- One way to perform Representation learning
- An approach to use **Self-supervised learning**

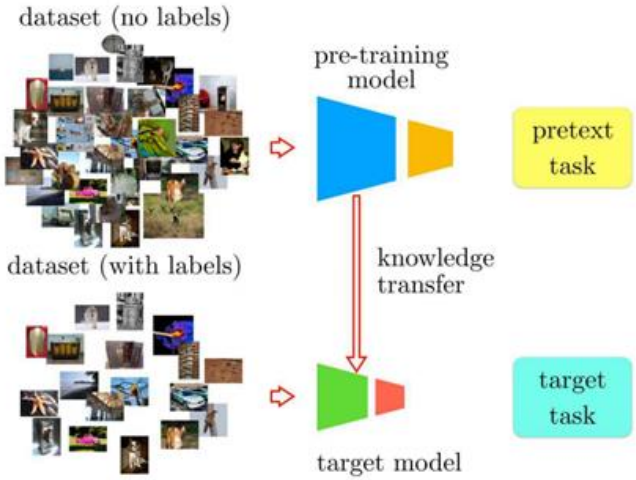
⇒ Learning through comparison between the input samples

  - Composed of **Positive** pair and **Negative** pair
  - Positive pair – close, Negative pair – far apart
  - To learn such an embedding space in which

    Similar samples stay **close** together, while **dissimilar** ones are **far** apart

- Advantages: data construction costs are none and learn easier
  - **Use Unlabelled data** ⇒ representation is more general, can respond to new classes
- It is used as a way to fine-tuning network for various Downstream tasks(classification)
  - It is much simpler to perform fine-tuning with other tasks in that it can be done without modification of the model structure





> Train with using Pretext task

https://daebaq27.tistory.com/97
https://velog.io/@jhlee508/논문-정리-A-Simple-Framework-for-Contrastive-Learning-of-Visual-Representations

A Simple Framework for Contrastive Learning of Visual Representations

Chen, Ting, et al. (Google Research)
International conference on machine learning. PMLR, 2020.

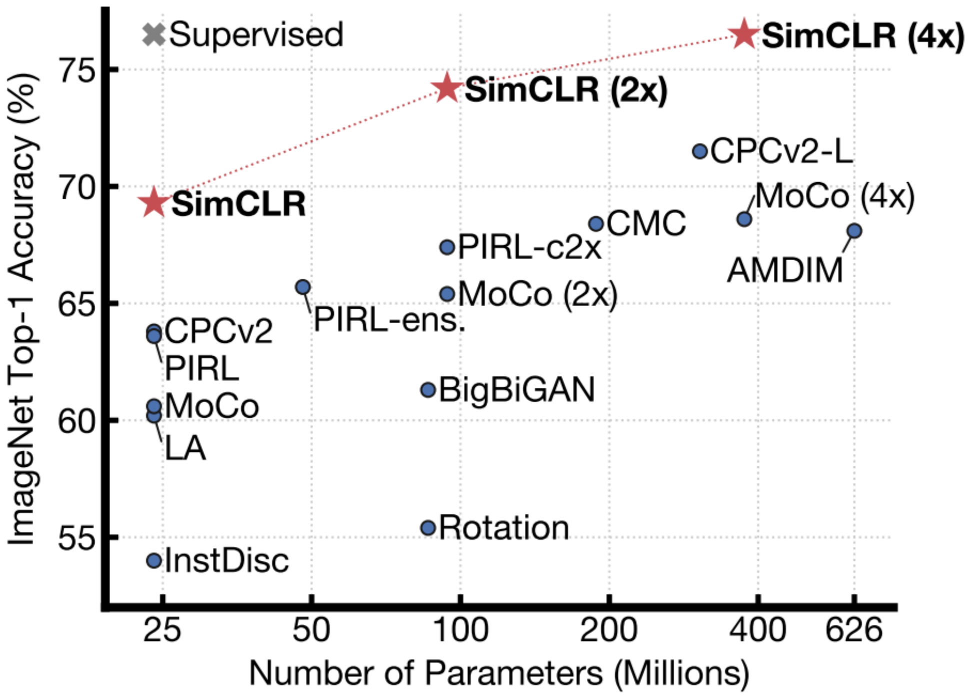
1. Introduction

❖ **SimCLR -** a simple framework for **contrastive learning** of visual representations

- Outperforms previous work
- Simpler, requiring neither specialized architectures nor a memory bank

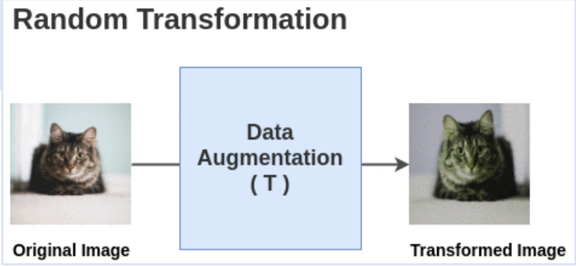
✓ Major components

1. Composition of multiple data augmentation operations is crucial in defining the contrastive prediction tasks
2. Introducing a learnable nonlinear transformation between the representation and the contrastive loss substantially improves the quality of the learned representations
3. Benefits from larger batch sizes and longer training compared to its supervised counterpart

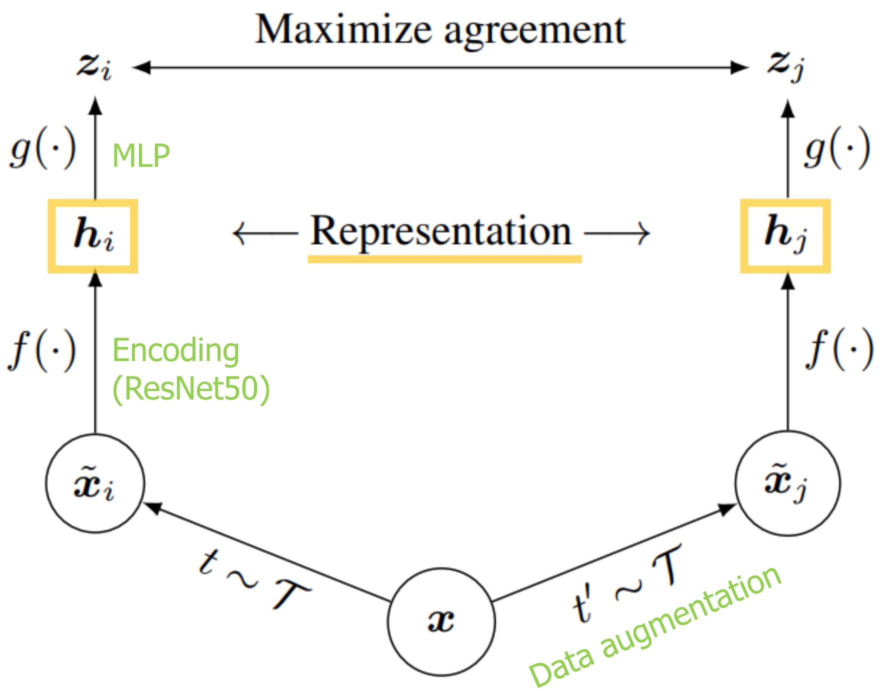


> ImageNet Top-1 accuracy of linear classifiers trained on representations learned with different self-supervised methods (pretrained on ImageNet). Gray cross indicates supervised ResNet-50. Our method, SimCLR, is shown in bold.

2. Method



❖ The Contrastive Learning Framework



- No explicit negative sampling
  - Minibatch(N) → 2N data points
  - **Negative** sample : **2(N-1)** data points
    Positive sample : **1** data point

★ **Data augmentation** module

- T : Data augmentations
  (Random Resize Crop, Random Color distortion, Gaussian Blur)
- t, t' : Two separate data augmentation operators
- x_i, x_j : Two correlated views

★ **Base Encoder $f(\cdot)$** : ResNet-50

- h_i, h_j : Representation vectors (Global Average Pooling)

★ **Projection head $g(\cdot)$** : map non-linear representation to the space

- Use Two-layer MLP
- z_i, z_j : vectors generated after Projection head

★ **Contrastive loss function** : NT-Xent Loss on z_i, z_j

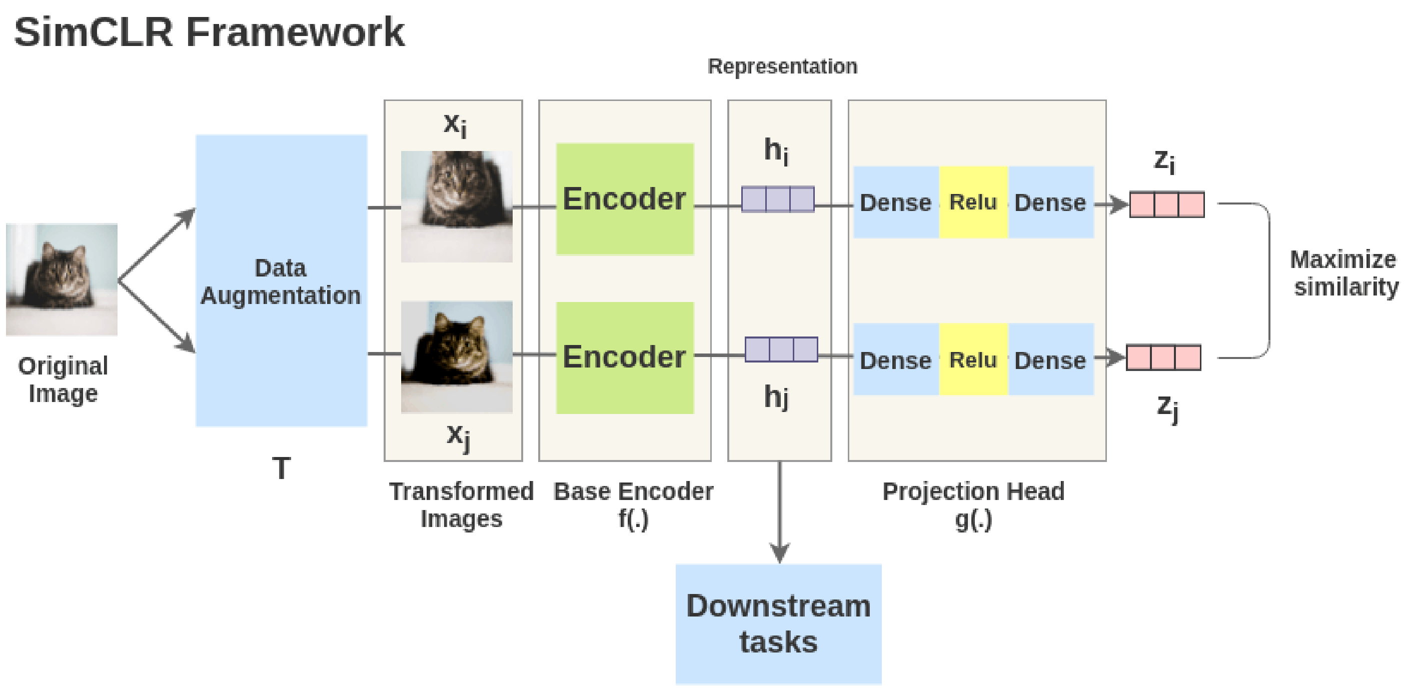
- Cosine similarity : $sim(u, v) = u^T v / ||u|| \, ||v||$

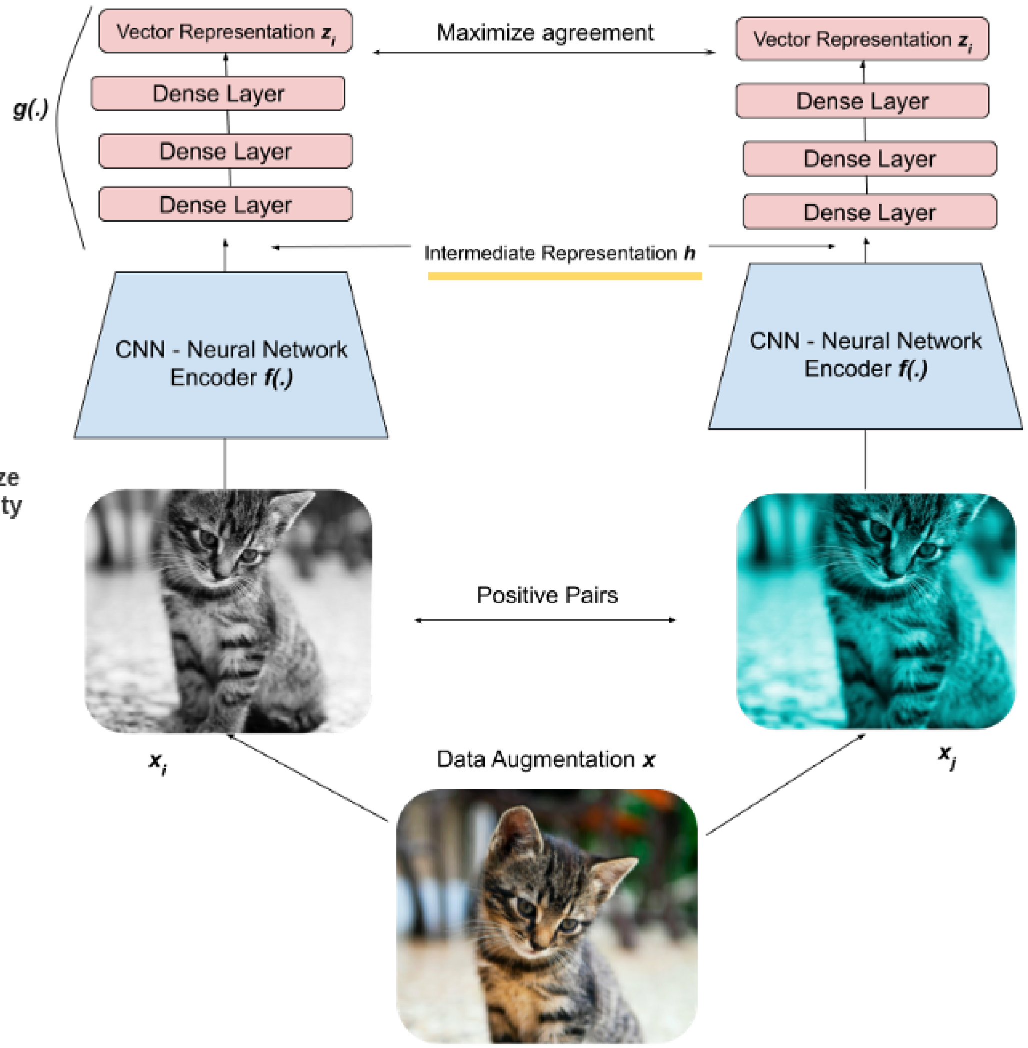
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)},$$

- ➢ Same image → hight similarity
- ➢ Different image → low similarity

2. Method

❖ The Contrastive Learning Framework overview



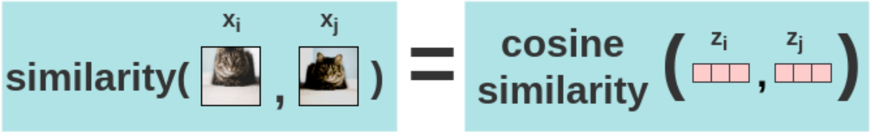


https://amitness.com/2020/03/illustrated-simclr/
https://towardsdatascience.com/understanding-contrastive-learning-d5b19fd96607

2. Method

❖ **Tuning Model: Bringing similar closer**

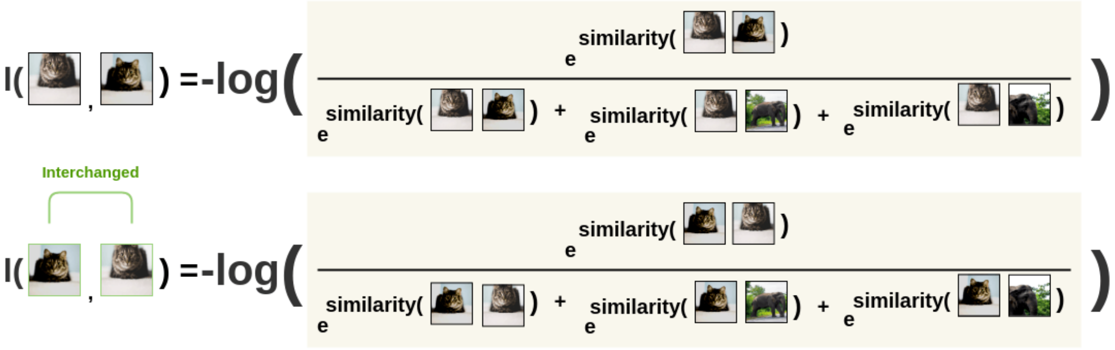
1. Calculation of **cosine similarity**



> Cosine similarity

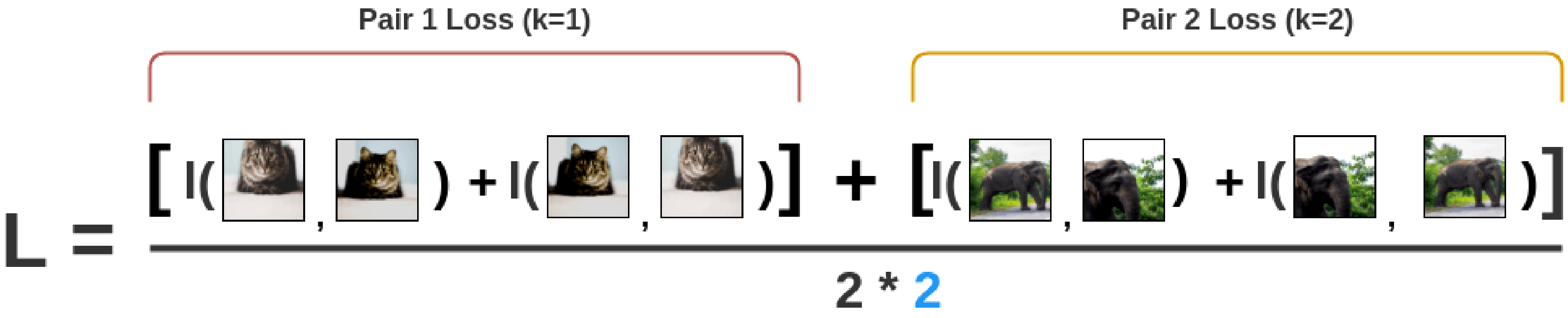
$$s_{i,j} = \frac{z_i^T z_j}{(\tau ||z_i|| ||z_j||)}$$

2. **NT-Xent Loss** calculation



> **NT-Xent Loss** : -log(softmax)

$$l(i,j) = -log \frac{exp(s_{i,j})}{\sum_{k=1}^{2N} l_{[k!=i]} exp(s_{i,k})}$$

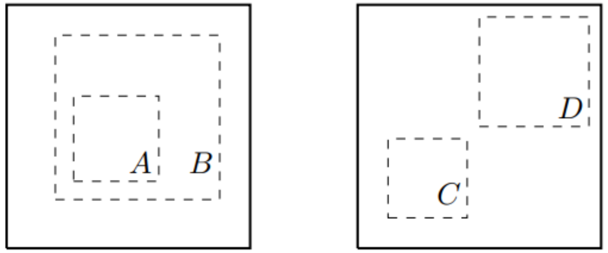


> Compute loss over all the pairs in the batch size (N=2) and take an average

$$L = \frac{1}{2N}\sum_{k=1}^{N}[l(2k-1, 2k) + l(2k, 2k-1)]$$

3. Data Augmentation for Contrastive Representation Learning

❖ Data Augmentation



(a) Global and local views. (b) Adjacent views.

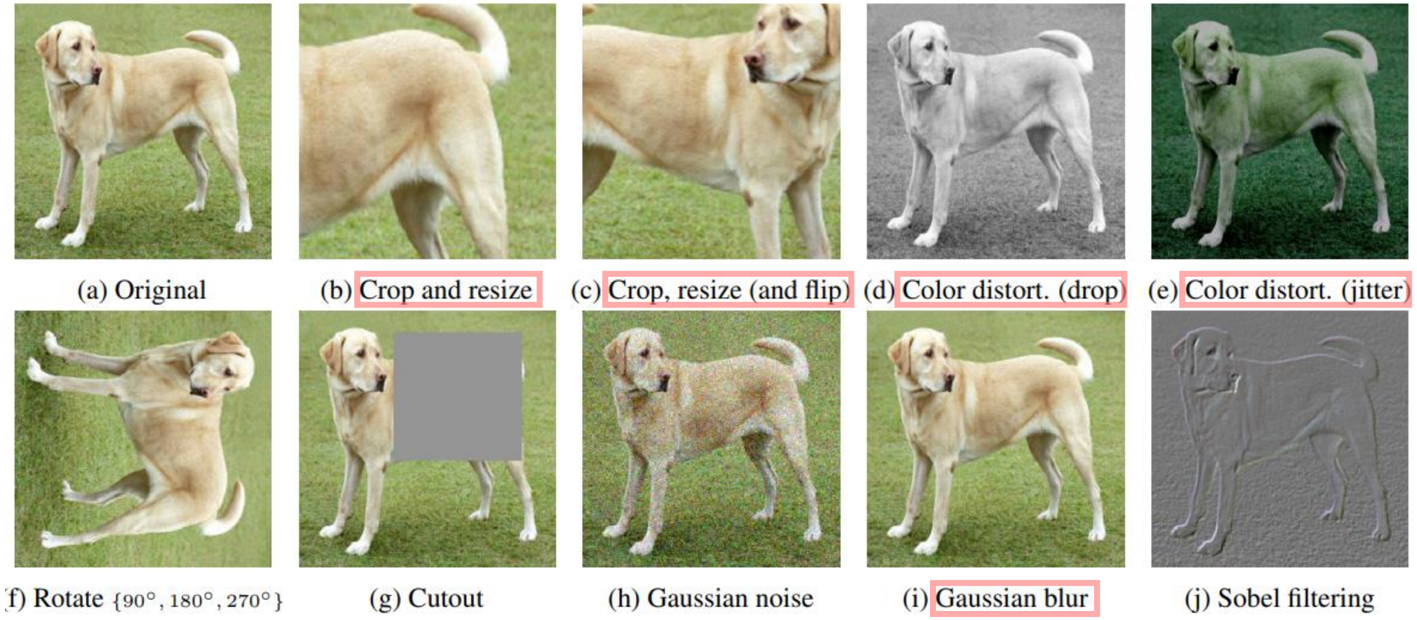
Figure 3. Solid rectangles are images, dashed rectangles are random crops. By randomly cropping images, we sample contrastive prediction tasks that include global to local view ($B \to A$) or adjacent view ($D \to C$) prediction.

- Many existing approaches :
  change the architecture to define contrastive prediction tasks

⇒ Perform simple **random cropping** of target images – contain all of things abocve

⇒ **Decouple**(분리) **Predictive task with NN architecture**

\> Augmentations

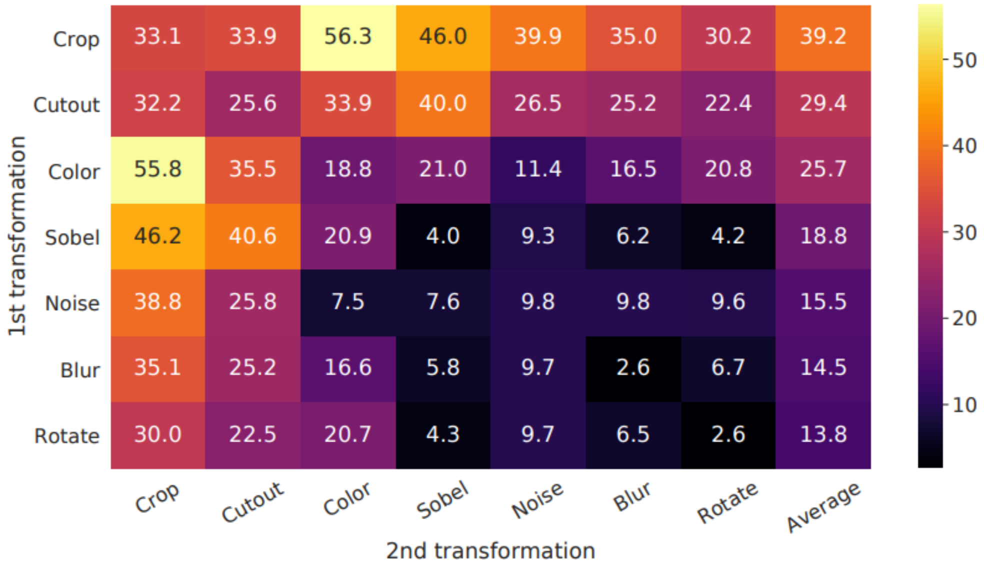


- Spatial/geometric transformation
  - Cropping, resizing, rotating, cutout
- Appearance transformation
  - Color distortion, Gaussian blur, Sobel filtering

Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

3. Data Augmentation for Contrastive Representation Learning

❖ Linear evaluation results under individual and composition of transformations



- No single transformation suffices to learn good representations
- When composing augmentations, the contrastive prediction task becomes harder, but the quality of representation improves
- Best : Random Crop + Random Color Distortion

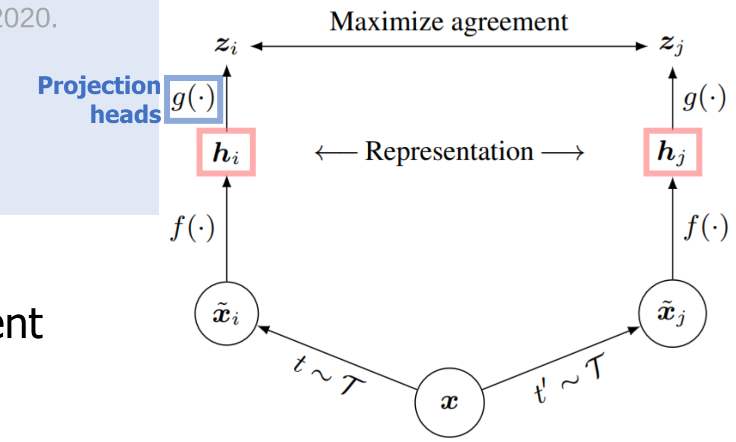
❖ Color Augmentation strength

Methods	Color distortion strength 1/8	1/4	1/2	1	1 (+Blur)	AutoAug
SimCLR	59.6	61.0	62.6	63.2	64.5	61.1
Supervised	77.0	76.7	76.5	75.7	75.4	77.1

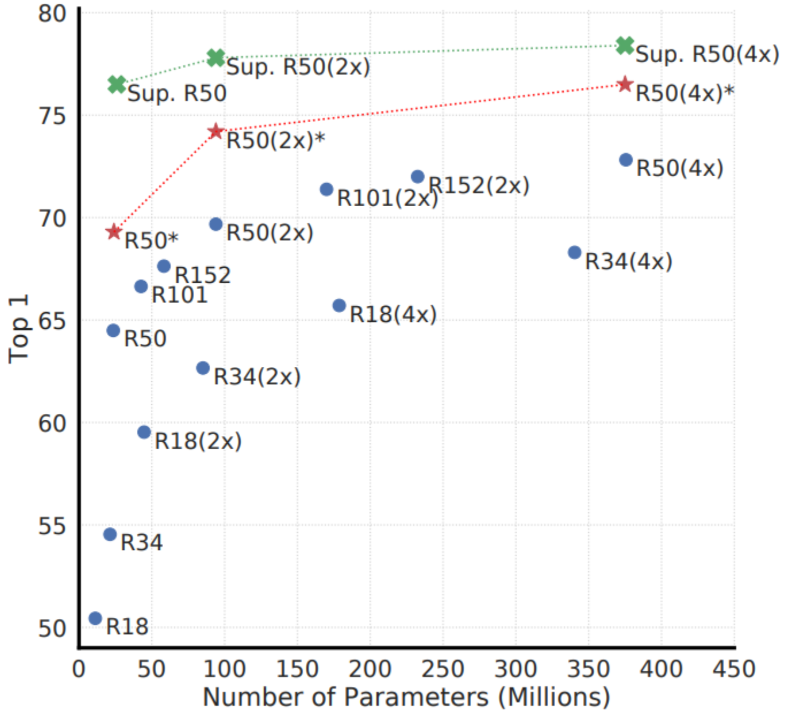
Table 1. Top-1 accuracy of unsupervised ResNet-50 using linear evaluation and supervised ResNet-50[5], under varied color distortion strength (see Appendix A) and other data transformations. Strength 1 (+Blur) is our default data augmentation policy.

- Stronger color augmentation → improves
- Simple cropping + stronger color distortion > Auto Augment(supervised)
- Unsupervised contrastive learning benefits from stronger (color) data augmentation than supervised learning

4. Architectures for Encoder and Head

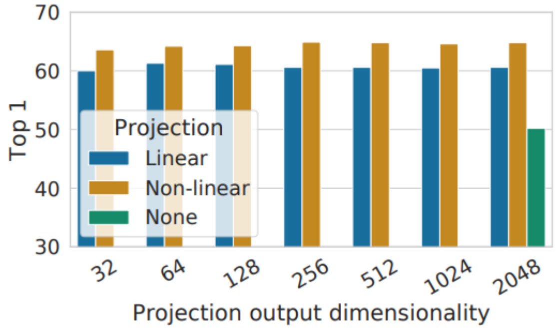


❖ Linear evaluation of **supervised / unsupervised**



- **Unsupervised** contrastive learning benefits (more) from bigger models

❖ Linear evaluation with different **projection head** $g(\cdot)$



- Projection :

Nonlinear > Linear > None

What to predict?	Random guess	Representation h	$g(h)$
Color vs grayscale	80	99.3	97.4
Rotation	25	67.6	25.6
Orig. vs corrupted	50	99.5	59.6
Orig. vs Sobel filtered	50	96.6	56.3

> Accuracy of training additional MLPs

- Representation = h – better than $g(h)$
  - $z = g(h)$: trained to be invariant to transformation
  - h – more information maintained / $g(h)$ - loses information

5. Loss Functions and Batch Size

❖ Loss function

Margin	NT-Logi.	Margin (sh)	NT-Logi.(sh)	NT-Xent
50.9	51.6	57.5	57.9	63.9

Table 4. Linear evaluation (top-1) for models trained with different loss functions. "sh" means using semi-hard negative mining.

ℓ_2 norm?	τ	Entropy	Contrastive acc.	Top 1
Yes	0.05	1.0	90.5	59.7
	0.1	4.5	87.8	64.4
	0.5	8.2	68.2	60.7
	1	8.3	59.1	58.0
No	10	0.5	91.7	57.2
	100	0.5	92.1	57.0

Table 5. Linear evaluation for models trained with different choices of ℓ_2 norm and temperature τ for NT-Xent loss. The contrastive distribution is over 4096 examples.

- NT-Xent performed best

❖ Batch size

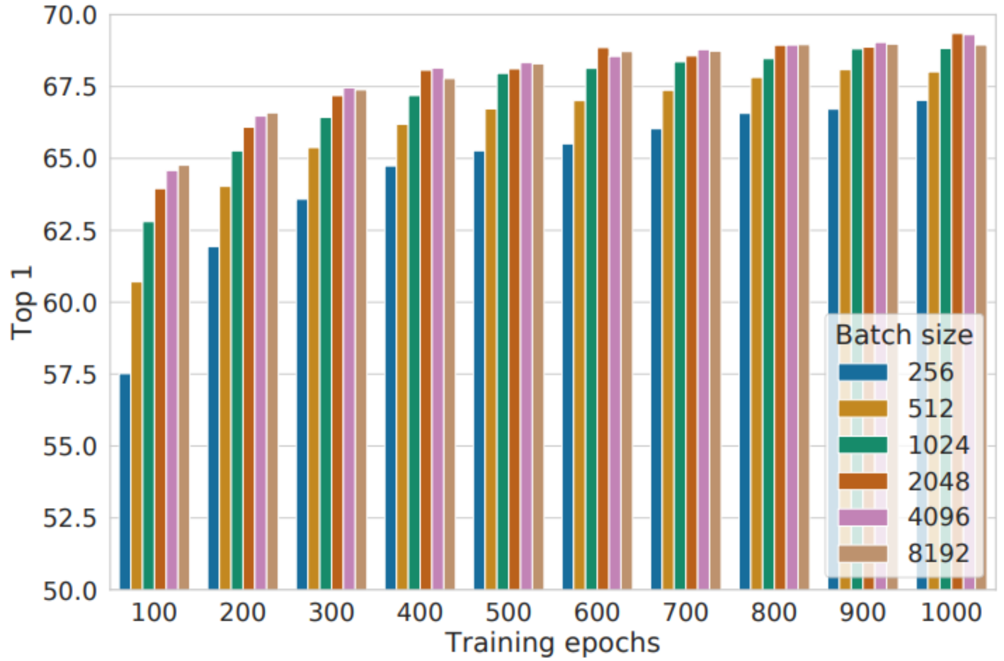
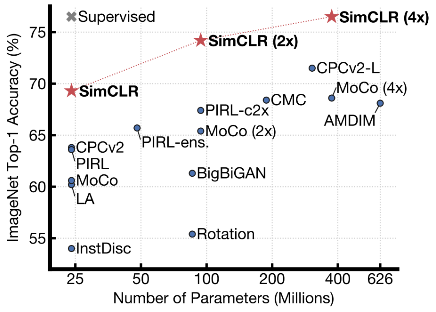


Figure 9. Linear evaluation models (ResNet-50) trained with different batch size and epochs. Each bar is a single run from scratch.[10]

- Larger Batch Size → better
  - Larger batch sizes provide more negative examples, facilitating convergence

Chen, Ting, et al. "A simple framework for contrastive learning of visual representations." International conference on machine learning. PMLR, 2020.

6. Comparison with State-of-the-art



- **Linear evaluation**

Method	Architecture	Param (M)	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

  - Compare with previous approaches (Self-supervised model)
  - **SimCLR** : best performance

- **Semi-supervised learning**

Method	Architecture	Label fraction 1%	10%
		Top 5	
Supervised baseline	ResNet-50	48.4	80.4
Methods using other label-propagation:			
Pseudo-label	ResNet-50	51.6	82.4
VAT+Entropy Min.	ResNet-50	47.0	83.4
UDA (w. RandAug)	ResNet-50	-	88.5
FixMatch (w. RandAug)	ResNet-50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet-50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet-50	39.2	77.4
BigBiGAN	RevNet-50 (4×)	55.2	78.8
PIRL	ResNet-50	57.2	83.8
CPC v2	ResNet-161(∗)	77.9	91.2
SimCLR (ours)	ResNet-50	75.5	87.8
SimCLR (ours)	ResNet-50 (2×)	83.0	91.2
SimCLR (ours)	ResNet-50 (4×)	**85.8**	**92.6**

Table 7. ImageNet accuracy of models trained with few labels.

- Sample 1% or 10% of the labeled ILSVRC-12 training datasets in a class-balanced way, fine-tune the whole base network on the labeled data

7. Conclusion

❖ SimCLR

- **Self-supervised Learning Simple framework**
  - Improve performance of Self-supervised learning, Semi-supervised learning, Transfer learning
- Difference from standard Supervised learning
  - **Data augmentation, non-linear projection head, the loss function**
- By training to learn Representation, performance achieved at the level of Supervised learning
- The strength of this simple framework suggests that, despite a recent surge in interest, self-supervised learning remains undervalued