Self-Supervised Representation Learning

Seokho Moon

May 1, 2020

논문, 블로그, 발표자료 및 다른 연구원분들(도형록, 강현구 등)의 도움을 받아 작성되었음을 미리 알려드리며 감사의 말을 전합니다.

https://lilianweng.github.io/lil-log/2019/11/10/self-supervised-learning.html

https://hoya012.github.io/blog/Self-Supervised-Learning-Overview/

PR-208: Unsupervised Visual Representation Learning Overview: Toward Self-Supervision

[Self-supervised learning] AI 프렌즈 세미나 발표자료 - 서정훈

Contents

1. Introduction
    1. Background
    2. Brief framework
2. Pretext tasks
    - Exemplar
    - Context Prediction(Relative Patch Location)
    - Jigsaw Puzzle
    - Image Colorization
    - Context Autoencoder
    - Count
    - Rotation
3. Downstream task evaluation
    - Transfer learning
4. SimCLR
    - Paper explanation
    - Results
5. Conclusion
    - Comments

Introduction

Background

딥러닝/머신러닝 모델을 잘 학습시키기 위해서는

양질의 데이터가 필요!

Introduction

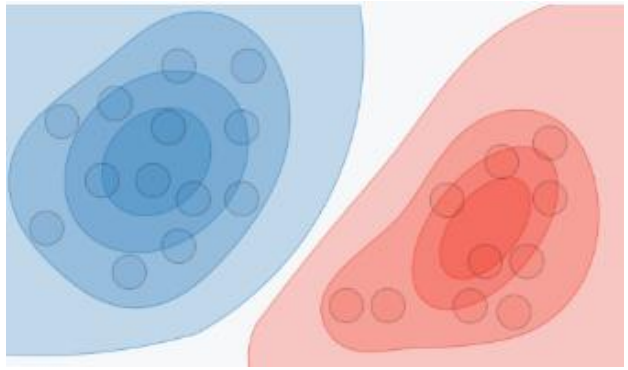
Background

딥러닝/머신러닝 모델을 잘 학습시키기 위해서는

양질의 데이터가 필요!

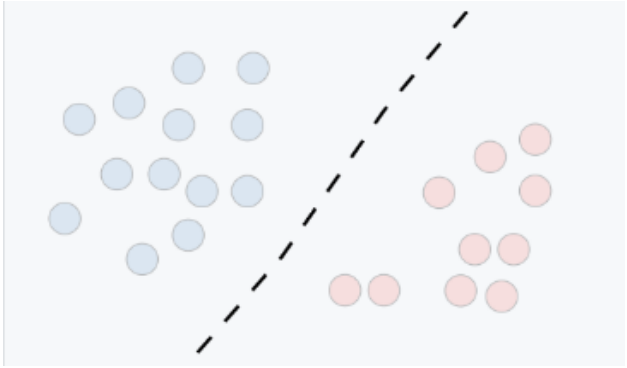
Generative model

→ Dataset의 분포를 잘 찾기 위해서



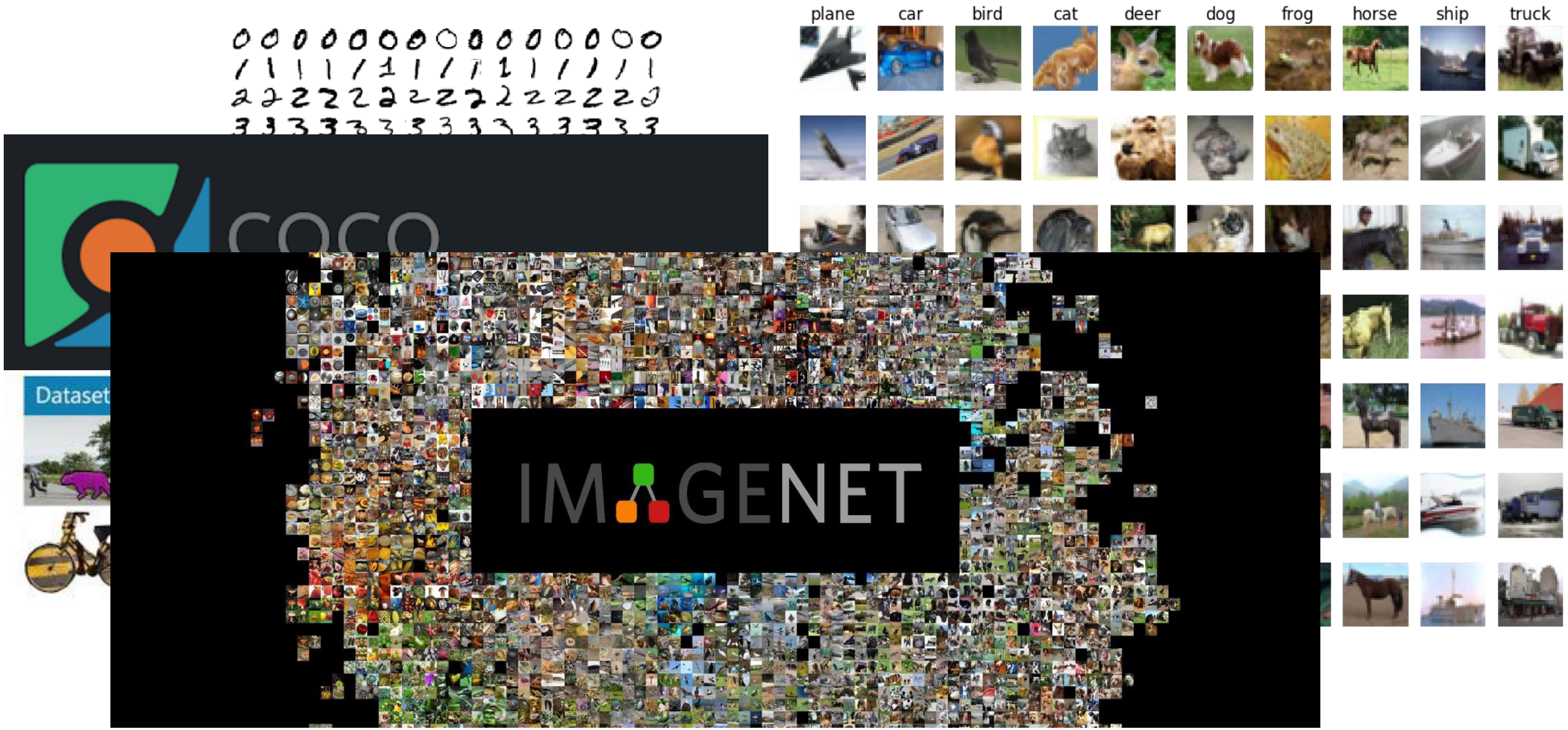
Discriminative model

→ Input과 output의 관계(함수)를 알아내기 위해서



Introduction

Background



ImageNet

Introduction

Background

양질의 데이터를 만들기 위해서는 많은 노력과 비용이 요구됨

Introduction

Background

양질의 데이터를 만들기 위해서는 많은 노력과 비용이 요구됨

이러한 문제에 도전하는 방법들

Transfer Learning

Domain Adaptation

Semi-Supervised learning

Weakly-supervised learning

Self-supervised learning

Introduction

Background

양질의 데이터를 만들기 위해서는 많은 노력과 비용이 요구됨

이러한 문제에 도전하는 방법들

Transfer Learning

Domain Adaptation

Semi-Supervised learning

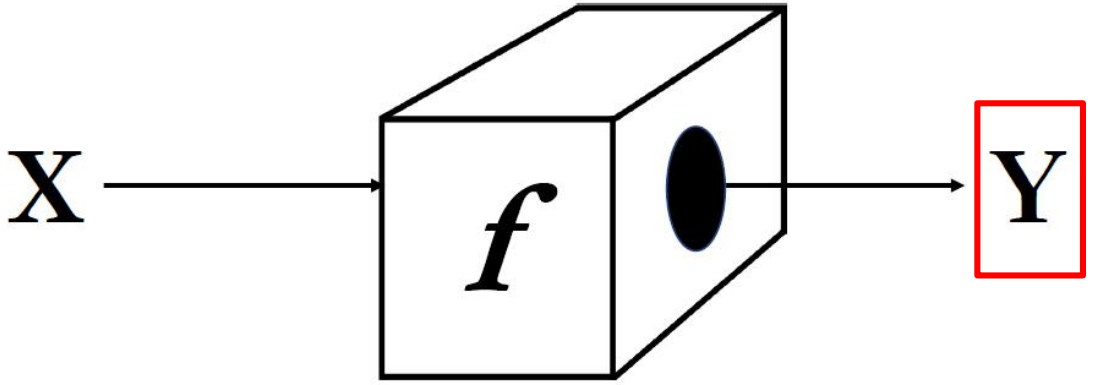
Weakly-supervised learning

Self-supervised learning → Unsupervised learning 방법론

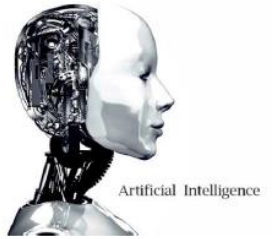
Introduction

Background

Supervision(지도, 감독)이란?



우리가 흔히 아는 label값이라고 생각하면 좋음



Introduction

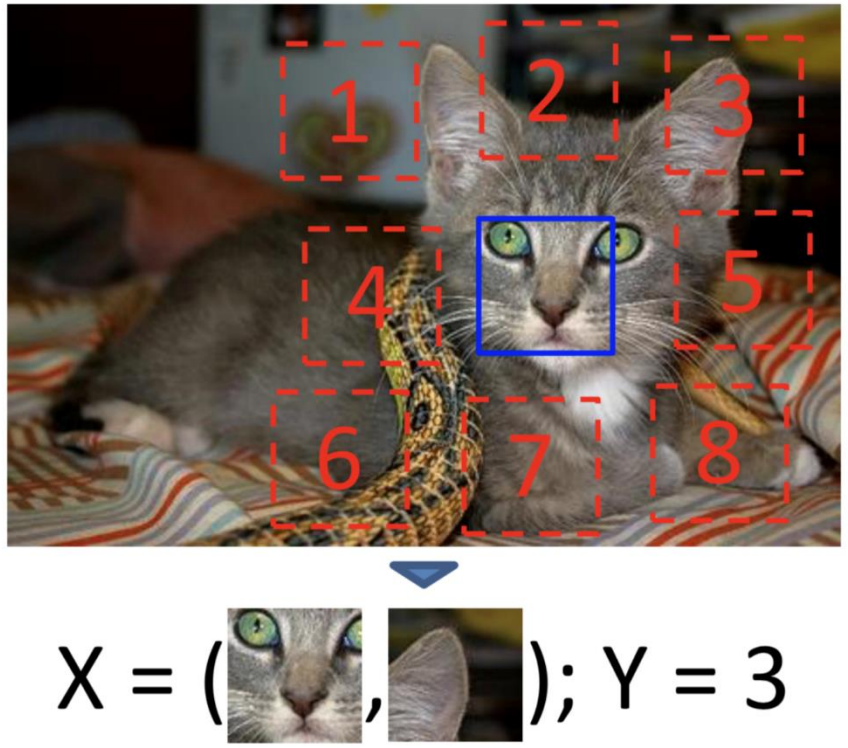
Background

Self-supervision 이란?

Image의 feature를 추출하는 모델이 있다고 할 때

Input 데이터의 한 부분이 다른 부분의 supervision

역할을 하는 것을 말함



Introduction

Background

인공지능 분야 세계적 석학 얀 르쿤(뉴욕대학교 교수)의 Facebook에서...

Yann LeCun

2019년 4월 30일 ·

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That's why calling it "unsupervised" is totally misleading. That's also why more knowledge about the structure of the world can be learned through self-supervised learning than from the other two paradigms: the data is unlimited, and amount of feedback provided by each example is huge.

Introduction

Background

인공지능 분야 세계적 석학 얀 르쿤(뉴욕대학교 교수)의 Facebook에서...

Yann LeCun

2019년 4월 30일 ·

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That's why calling it "unsupervised" is totally misleading. That's also why more knowledge about the structure of the world can be learned through self-supervised learning than from the other two paradigms: the data is unlimited, and amount of feedback provided by each example is huge.

Introduction

Background

1993년 LeNet1 테스트 영상

Introduction

Background

Yann LeCun
2019년 4월 30일 ·

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That's why calling it "unsupervised" is totally misleading. That's also why more knowledge about the structure of the world can be learned through self-supervised learning than from the other two paradigms: the data is unlimited, and amount of feedback provided by each example is huge.

"Self-supervised learning" 이라고 하겠다

"Unsupervised" 는 많이 포괄적인 의미를 지니고 헷갈리는 용어이다

Introduction

Background

Yann LeCun
2019년 4월 30일 ·

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That's why calling it "unsupervised" is totally misleading. That's also why more knowledge about the structure of the world can be learned through self-supervised learning than from the other two paradigms: the data is unlimited, and amount of feedback provided by each example is huge.

Self-supervised learning 에서는 입력 값의 한 부분으로 입력 값의 다른 부분을 예측하게 한다

Introduction

Background

Yann LeCun
2019년 4월 30일 ·

I now call it "self-supervised learning", because "unsupervised" is both a loaded and confusing term.

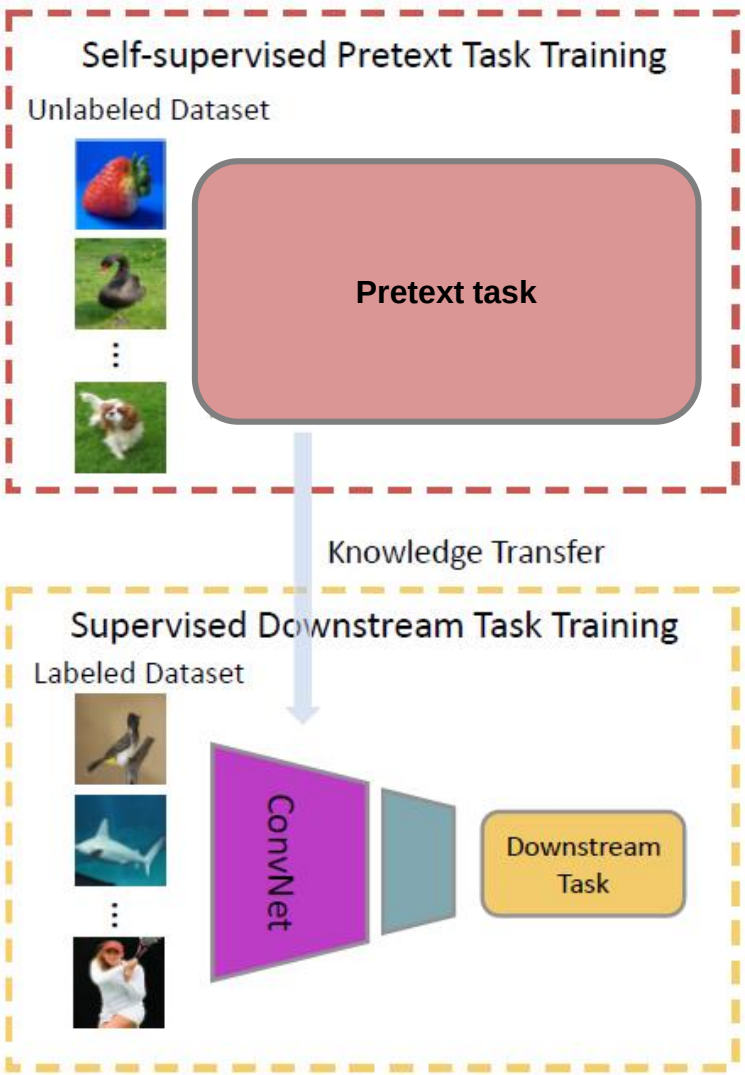
In self-supervised learning, the system learns to predict part of its input from other parts of it input. In other words a portion of the input is used as a supervisory signal to a predictor fed with the remaining portion of the input.

Self-supervised learning uses way more supervisory signals than supervised learning, and enormously more than reinforcement learning. That's why calling it "unsupervised" is totally misleading. That's also why more knowledge about the structure of the world can be learned through self-supervised learning than from the other two paradigms: the data is unlimited, and amount of feedback provided by each example is huge.

입력 값의 자체로 supervision을 만들어서 학습하기 때문에 “unsupervised” 라는 용어는 잘못되었다

Introduction

Brief framework

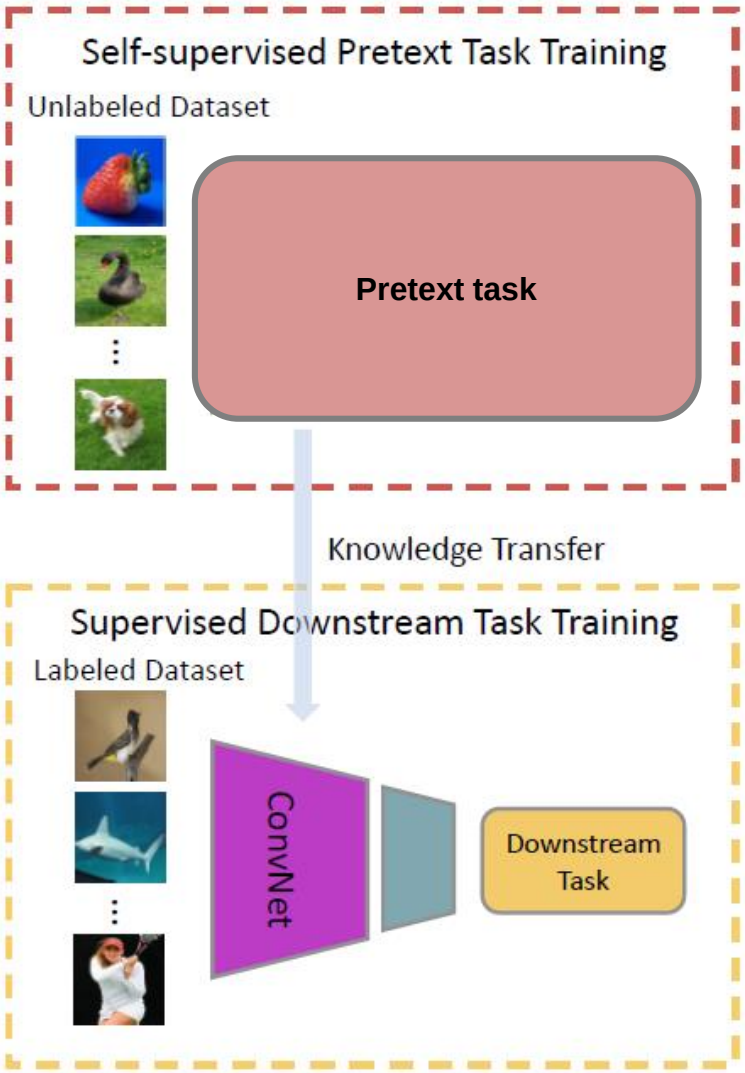


Unlabeled dataset를 input으로 받아 사용자가 정의한 문제(pretext task)를 network가 학습하게 하여 데이터 자체에 대한 이해도를 높이고자 함

Pretext task가 잘 짜여졌다면 input을 효과적으로 representation할 것이라는 가정이 존재

Introduction

Brief framework



Pretext task에서 나온 pre-training 되어진 network를 궁극적으로 사용자가 풀고자 하는 문제 (downstream task, e.g. classification)에 transfer learning하는 방법

Pretext tasks

Discriminative unsupervised feature learning with exemplar convolutional neural networks

❖ Exemplar, 2014 NIPS

- Pretext task 과정
  1. 기존의 96 x 96 size 이미지에서 object가 있을만한 곳을 crop하여 32 x 32 size의 seed patch를 얻음
  2. Seed patch에서 augmentation 기법을 통해 24개의 dataset 확보
  3. Classifier가 seed patch로부터 늘어난 patch들을 동일한 class로 예측하도록 학습 진행

- 한계점
  1. self-supervised representation learning 초기 연구 단계여서 dataset이 커질 경우 적당한 대응방법이 없음(class의 수에 맞게 classifier가 전부 학습해야 함)

Seed patch →

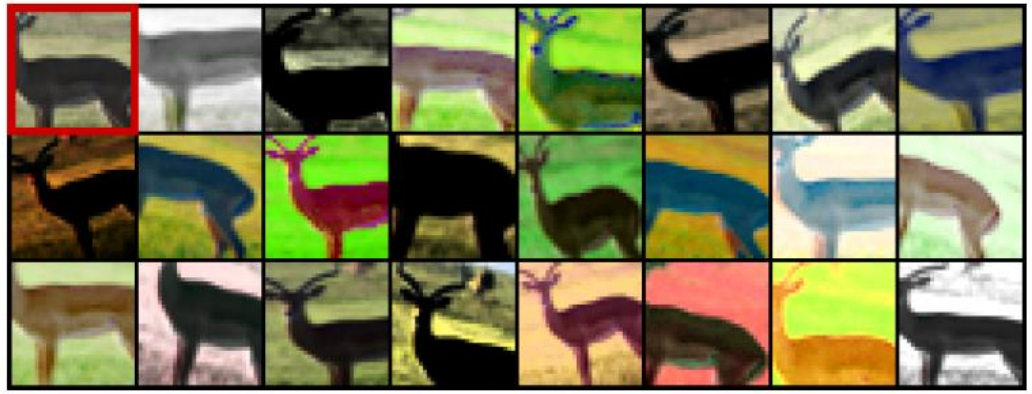


Fig. 2. Several random transformations applied to one of the patches extracted from the STL unlabeled dataset. The original ('seed') patch is in the top left corner.

Pretext tasks

Discriminative unsupervised feature learning with exemplar convolutional neural networks

❖ Exemplar, 2014 NIPS

- 성능

1. Classification accuracy가 그 당시에 존재했던 방법과 비교하여 우수함을 보임

TABLE 1
Classification accuracies on several datasets (in percent). ∗ Average per-class accuracy[1] $78.0\% \pm 0.4\%$. † Average per-class accuracy $85.0\% \pm 0.7\%$. ‡ Average per-class accuracy $85.8\% \pm 0.7\%$.

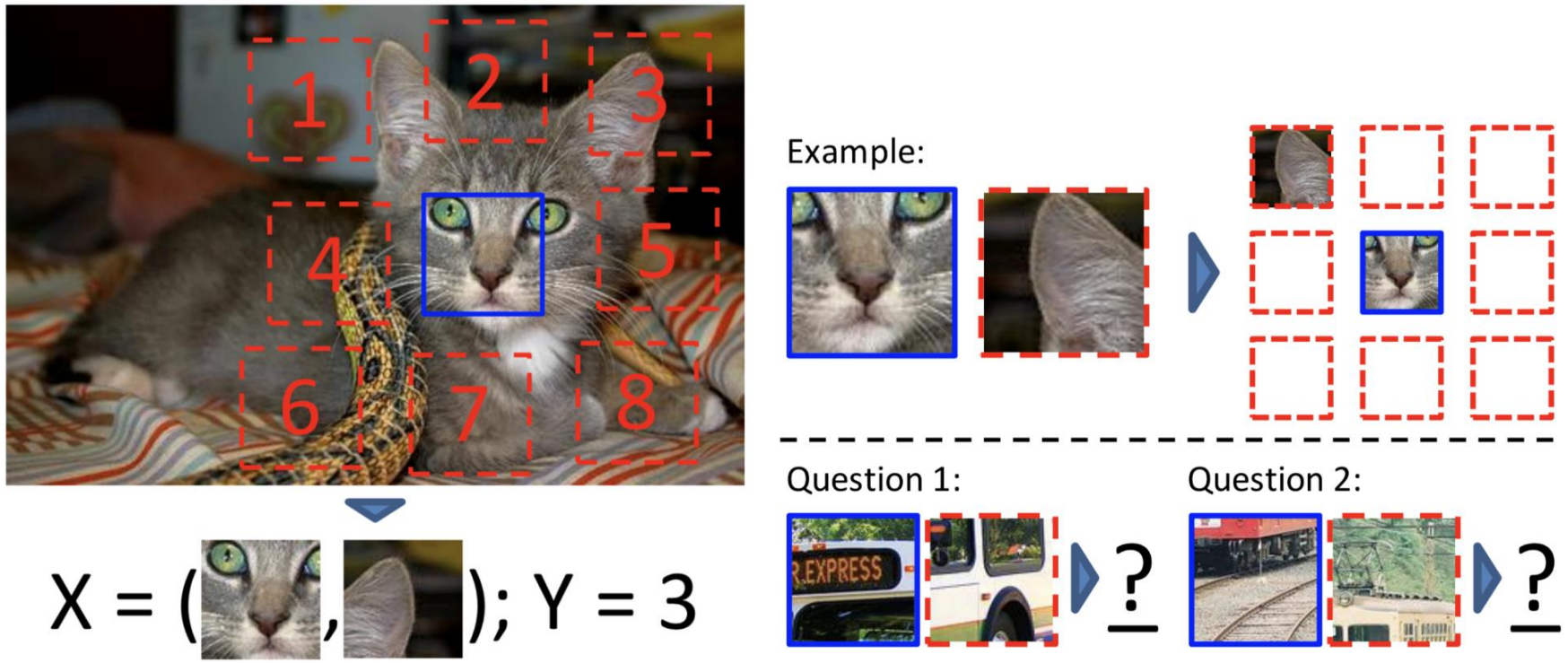
Algorithm	STL-10	CIFAR-10(400)	CIFAR-10	Caltech-101	Caltech-256(30)	#features
Convolutional K-means Network [32]	60.1 ± 1	70.7 ± 0.7	82.0	—	—	8000
Multi-way local pooling [33]	—	—	—	77.3 ± 0.6	41.7	1024×64
Slowness on videos [14]	61.0	—	—	74.6	—	556
Hierarchical Matching Pursuit (HMP) [34]	64.5 ± 1	—	—	—	—	1000
Multipath HMP [35]	—	—	—	82.5 ± 0.5	50.7	5000
View-Invariant K-means [16]	63.7	72.6 ± 0.7	81.9	—	—	6400
Exemplar-CNN (64c5-64c5-128f)	67.1 ± 0.2	69.7 ± 0.3	76.5	$79.8 \pm 0.5^{*}$	42.4 ± 0.3	256
Exemplar-CNN (64c5-128c5-256c5-512f)	72.8 ± 0.4	75.4 ± 0.2	82.2	$86.1 \pm 0.5^{\dagger}$	51.2 ± 0.2	960
Exemplar-CNN (92c5-256c5-512c5-1024f)	$\mathbf{74.2 \pm 0.4}$	$\mathbf{76.6 \pm 0.2}$	**84.3**	$\mathbf{87.1 \pm 0.7}^{\ddagger}$	$\mathbf{53.6 \pm 0.2}$	1884
Supervised state of the art	70.1[36]	—	92.0 [37]	91.44 [38]	70.6 [2]	—

Pretext tasks

Unsupervised Visual Representation Learning by Context Prediction

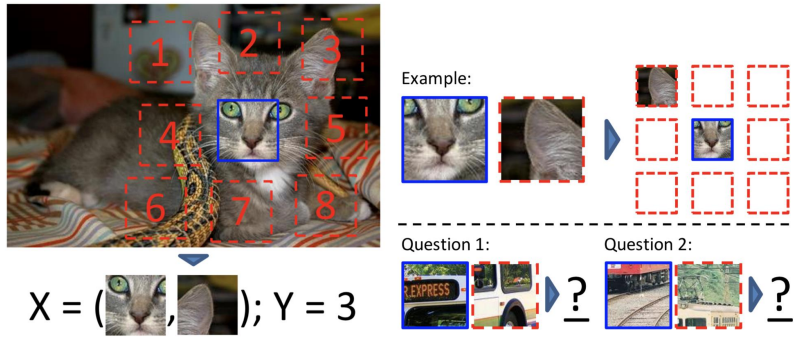
❖ Context Prediction(Relative Patch Location), 2015 ICCV

- Pretext task 과정



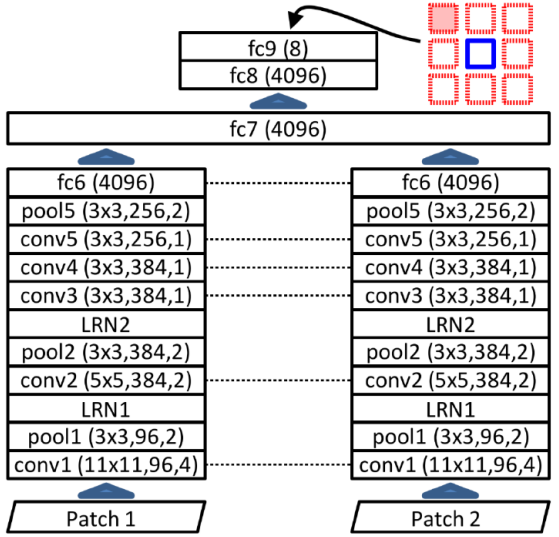
Pretext tasks

Unsupervised Visual Representation Learning by Context Prediction



❖ Context Prediction(Relative Patch Location), 2015 ICCV

- Pretext task 과정
  1. Input 이미지 1장에서 9개의 patch를 만들어 낸 후에 가운데 patch와 다른 patch와의 위치 정보를 classifier 에 학습시킴
  2. 사람이 단번에 알아채기 어려울 정도이므로(question 1, question 2) input 이미지를 잘 이해할 것이라 기대
  3. AlexNet에 pair한 input patch를 넣어주는 구조
  4. Trivial solution들을 막기 위한 방안
     - Patch간에 거리를 둠
     - Patch 사이의 거리가 조금씩 다름



Pretext tasks

Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles

❖ Jigsaw Puzzle, 2016 ECCV

- Pretext task 과정

(a) (b) (c)

Fig. 1: Learning image representations by solving Jigsaw puzzles. (a) The image from which the tiles (marked with green lines) are extracted. (b) A puzzle obtained by shuffling the tiles. Some tiles might be directly identifiable as object parts, but others are ambiguous (*e.g.*, have similar patterns) and their identification is much more reliable when all tiles are jointly evaluated. In contrast, with reference to (c), determining the relative position between the central tile and the top two tiles from the left can be very challenging [10].

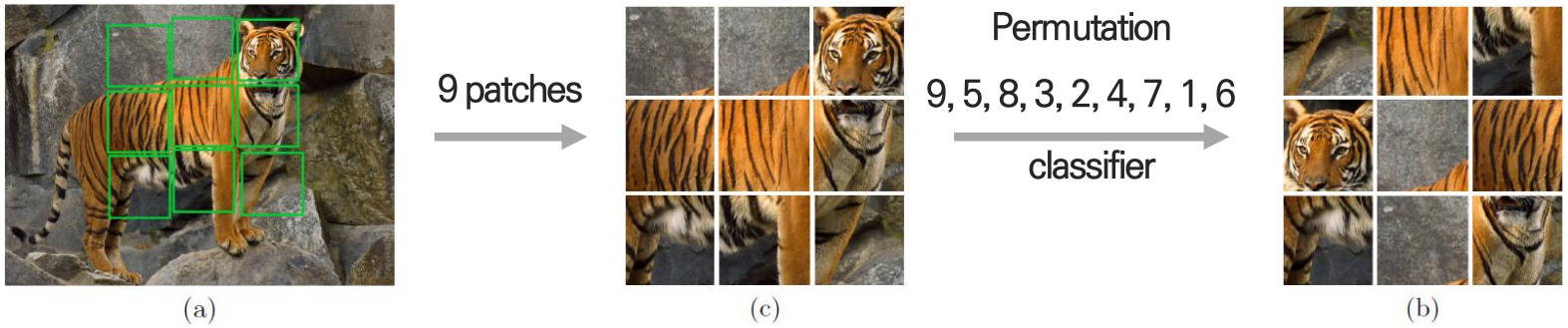
Pretext tasks

Unsupervised Learning of Visual Representations by Solving Jigsaw Puzzles

❖ Jigsaw Puzzle, 2016 ECCV

- Pretext task 과정

1. Input 이미지로부터 9개의 patch들을 만들어 냄
2. Patch들을 섞은 다음에 원래의 배치로 돌아가기 위한 순열을 예측하는 classifier를 학습



3. 9개의 patch들로 만들 수 있는 순열 조합은 9!(약 35만 개)이므로 비슷한 순열끼리 제거하여 대표적인 100개의 순열만 class로 분류하도록 학습

Pretext tasks

Colorful Image Colorization

❖ Image Colorization, 2016 ECCV

- Pretext task 과정

1. 1개의 channel(Y)만 가지고 있는 input 이미지에서 3 channel(Y,U,V)의 color 이미지를 생성하는 것을 학습시키고, 이때 사용한 encode를 기존 방식처럼 downstream task에 transfer하는 방법을 사용

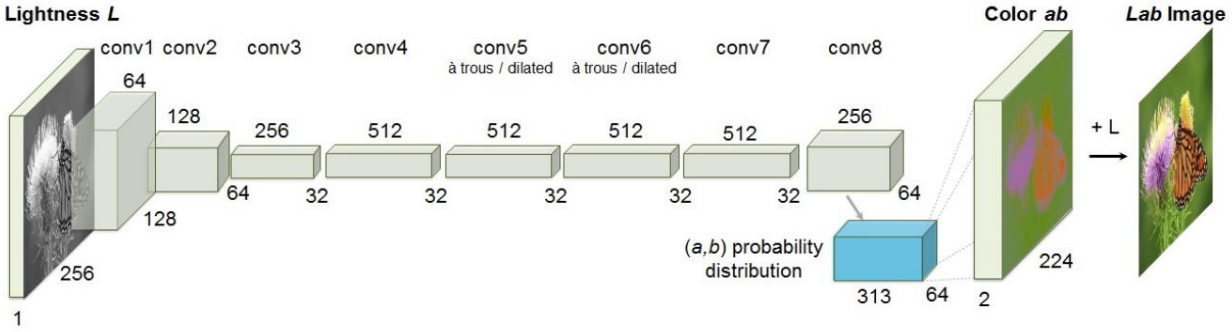


Fig. 2. Our network architecture. Each `conv` layer refers to a block of 2 or 3 repeated `conv` and `ReLU` layers, followed by a `BatchNorm` [30] layer. The net has no `pool` layers. All changes in resolution are achieved through spatial downsampling or upsampling between `conv` blocks.

Fig. 8. Applying our method to legacy black and white photos. Left to right: photo by David Fleay of a Thylacine, now extinct, 1936; photo by Ansel Adams of Yosemite; amateur family photo from 1956; *Migrant Mother* by Dorothea Lange, 1936.

Pretext tasks

Context Encoders: Feature Learning by Inpainting

❖ Context Autoencoder, 2016 CVPR

- Pretext task 과정
  1. 중간 부분이 잘린 이미지를 context 이미지라고 하여 autoencoder를 통과시키면, missing regions를 예측하고 실제 잘려나간 부분과 비교해가면서 학습 진행
  2. 마찬가지로 필요한 부분을 downstream task에 transfer하는 방식을 사용

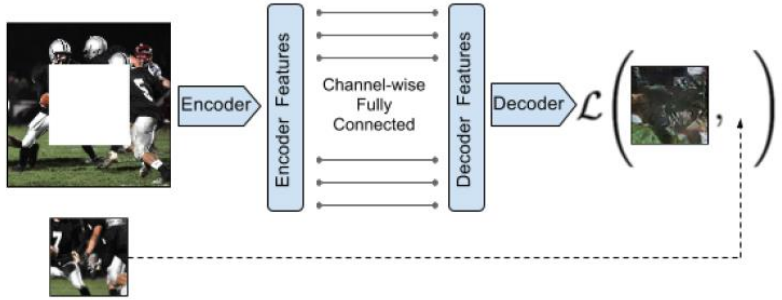
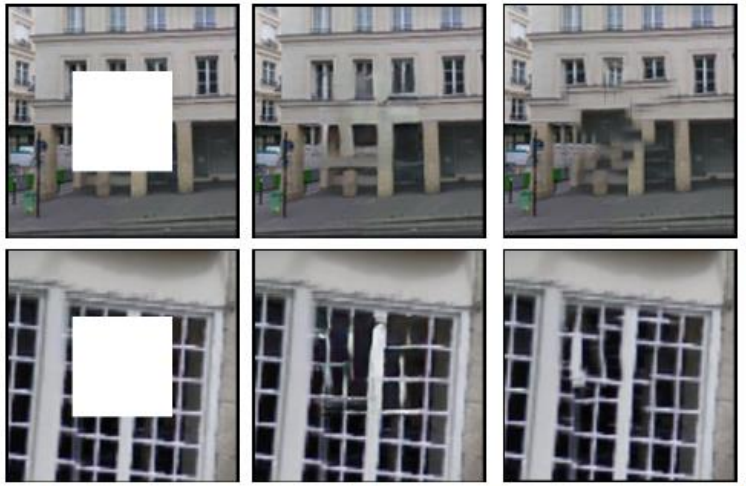


Figure 2: Context Encoder. The context image is passed through the encoder to obtain features which are connected to the decoder using channel-wise fully-connected layer as described in Section 3.1. The decoder then produces the missing regions in the image.



Input Context | Context Encoder | Content-Aware Fill

Pretext tasks

Representation Learning by Learning to Count

❖ Count, 2017 ICCV

- Pretext task 과정
  1. 앞선 방식들은 순서를 바꾸거나 이미지의 일부를 자른 후 다시 복원하는 형태
  2. 이번에는 이미지로부터 임의의 특징값을 vector로 뽑아내어 사용
     - Nose, eyes, paws, head
     - 이 값들은 쪼개어져도 각 patch에서 특징값의 합이 전체 특징값과 일치함

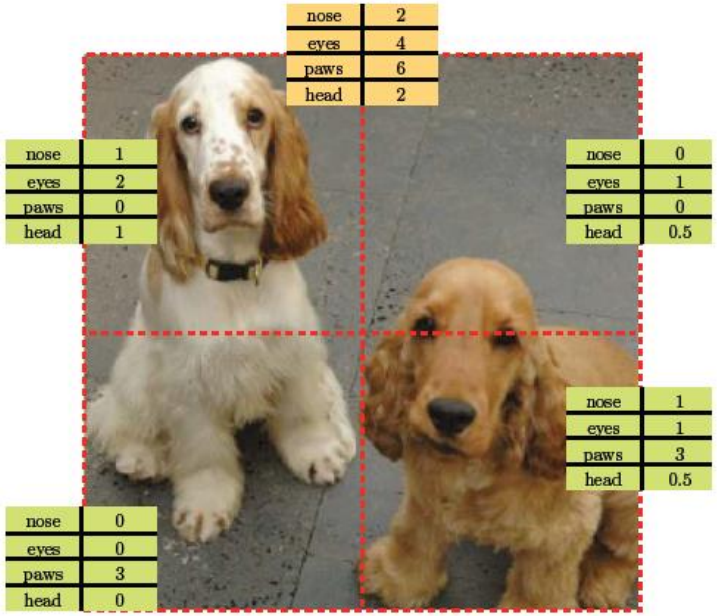


Figure 1: The number of visual primitives in the whole image should match the sum of the number of visual primitives in each tile (dashed red boxes).

Pretext tasks

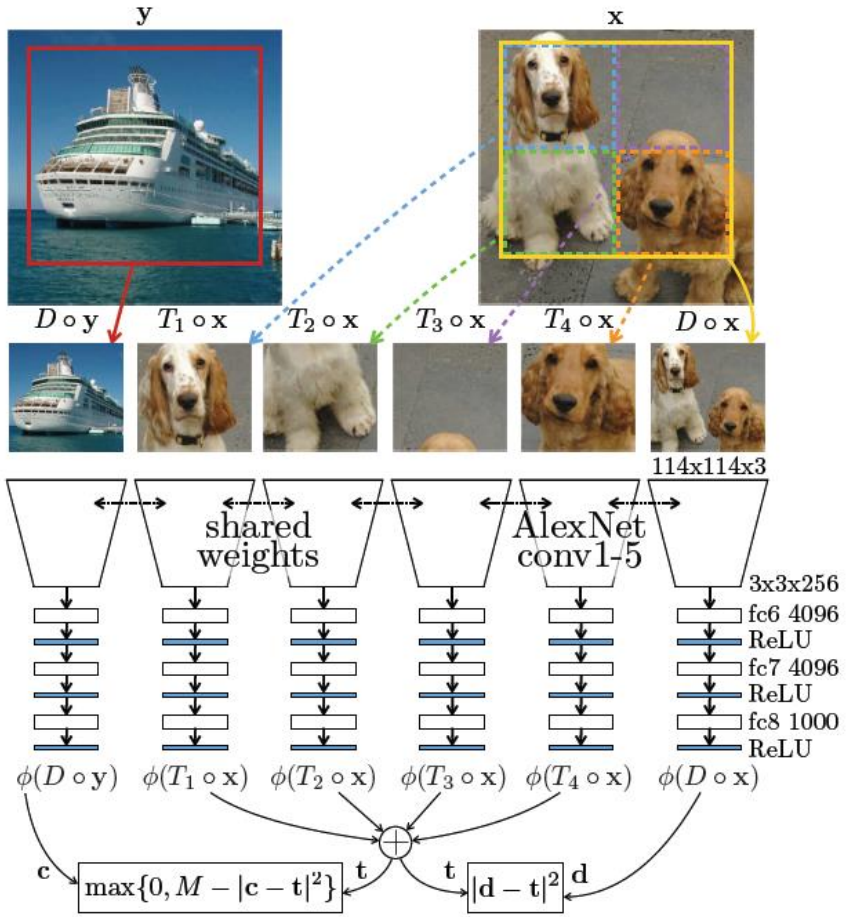
Representation Learning by Learning to Count

❖ Count, 2017 ICCV

- Pretext task 과정

3. Input 이미지를 down-sampling(D)한 것과 이것을 다시 tiling(T)한 것으로 나누고, 앞서 말한 것처럼 특징값의 합이 같도록 학습

$$\ell_{\text{con}}(\mathbf{x}, \mathbf{y}) = \left|\phi(D \circ \mathbf{x}) - \sum_{j=1}^{4} \phi(T_j \circ \mathbf{x})\right|^2 + \max\left\{0, M - \left|\phi(D \circ \mathbf{y}) - \sum_{j=1}^{4} \phi(T_j \circ \mathbf{x})\right|^2\right\}$$



➢ Feature $\phi : \mathbb{R}^{p \times q \times 3} \mapsto \mathbb{R}^k$ 가 잘 학습된다는 것이 앞서 말한 이미지를 적절한 특징값으로 잘 뽑아낸다는 것이므로 이것을 loss로 설정

4. Trivial solution을 방지하기 위해 전혀 다른 값 y를 넣어서 contrastive term을 추가함

Pretext tasks

Unsupervised representation learning by predicting image rotations

❖ Rotation, 2018 ICLR

- Pretext task 과정

1. Input 이미지에 0도, 90도, 180도, 270도 회전하여 나온 이미지를 input 이미지를 기반으로 rotation을 예측하는 4 class classification 문제

2. 전체 네트워크 안에 ConvNet이 들어있는 Network-In-Network(NIN) 방식을 사용한 RotNet 구조를 본 논문에서 제안함

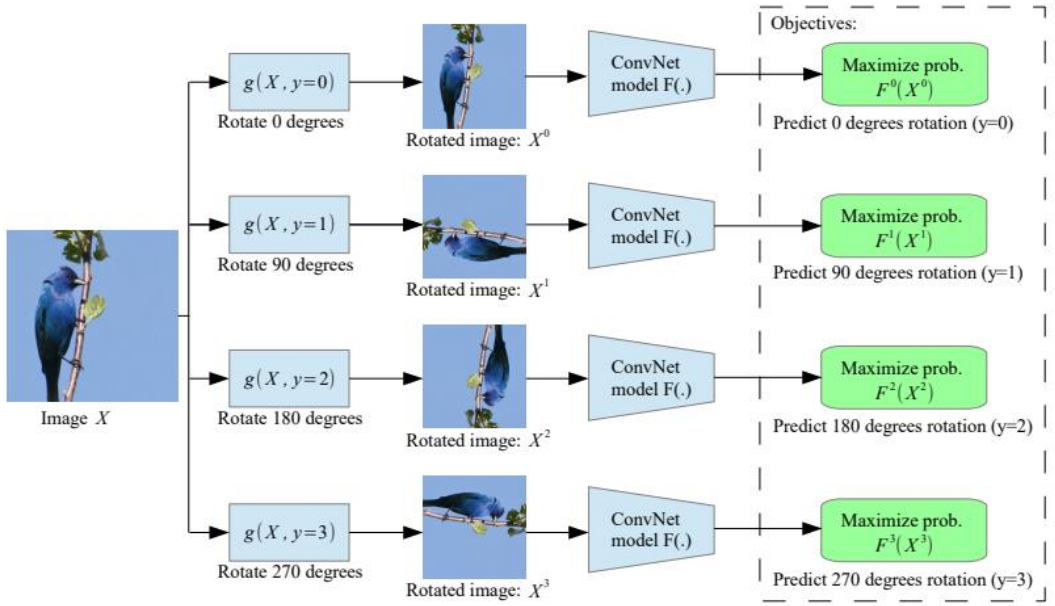
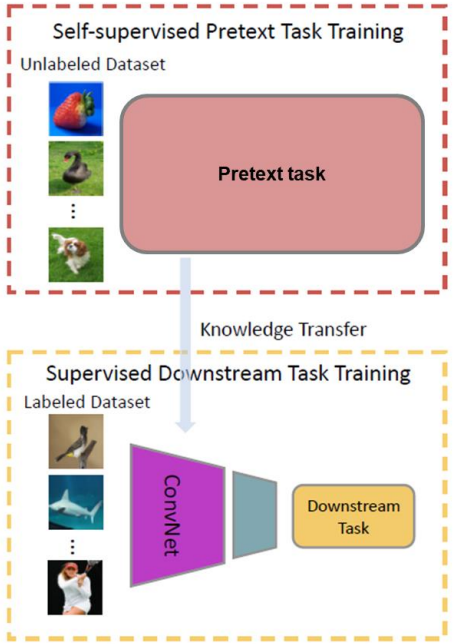


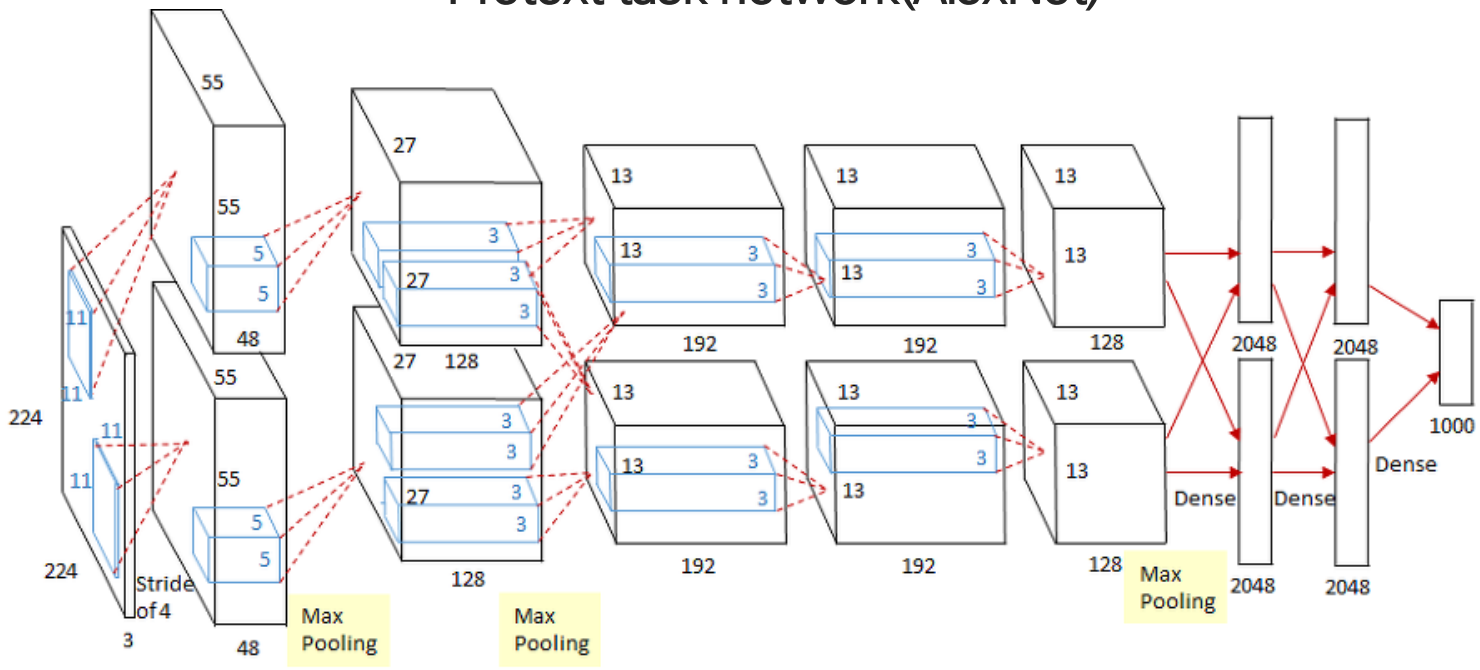
Figure 2: Illustration of the self-supervised task that we propose for semantic feature learning. Given four possible geometric transformations, the 0, 90, 180, and 270 degrees rotations, we train a ConvNet model $F(.)$ to recognize the rotation that is applied to the image that it gets as input. $F^y(X^{y^*})$ is the probability of rotation transformation y predicted by model $F(.)$ when it gets as input an image that has been transformed by the rotation transformation y^*.

Downstream task evaluation

Transfer learning



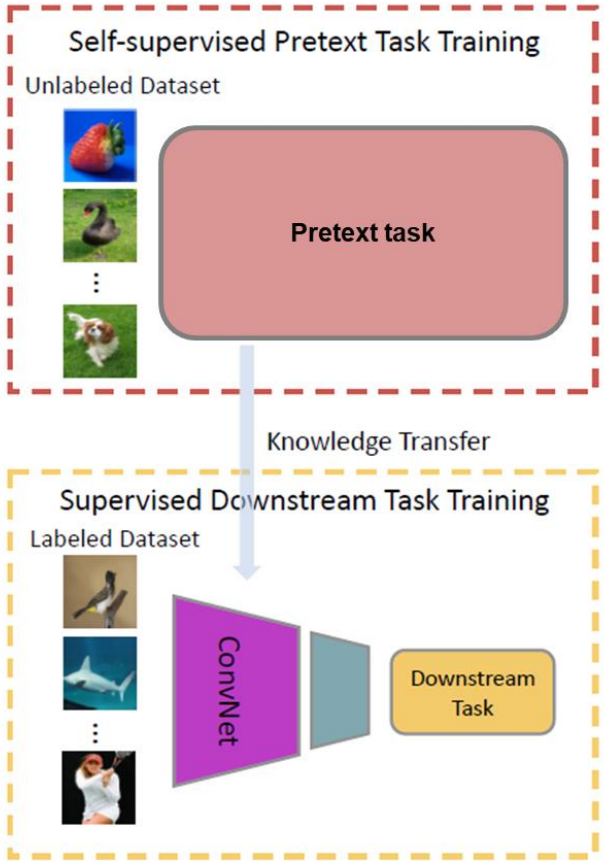
Pretext task network(AlexNet)



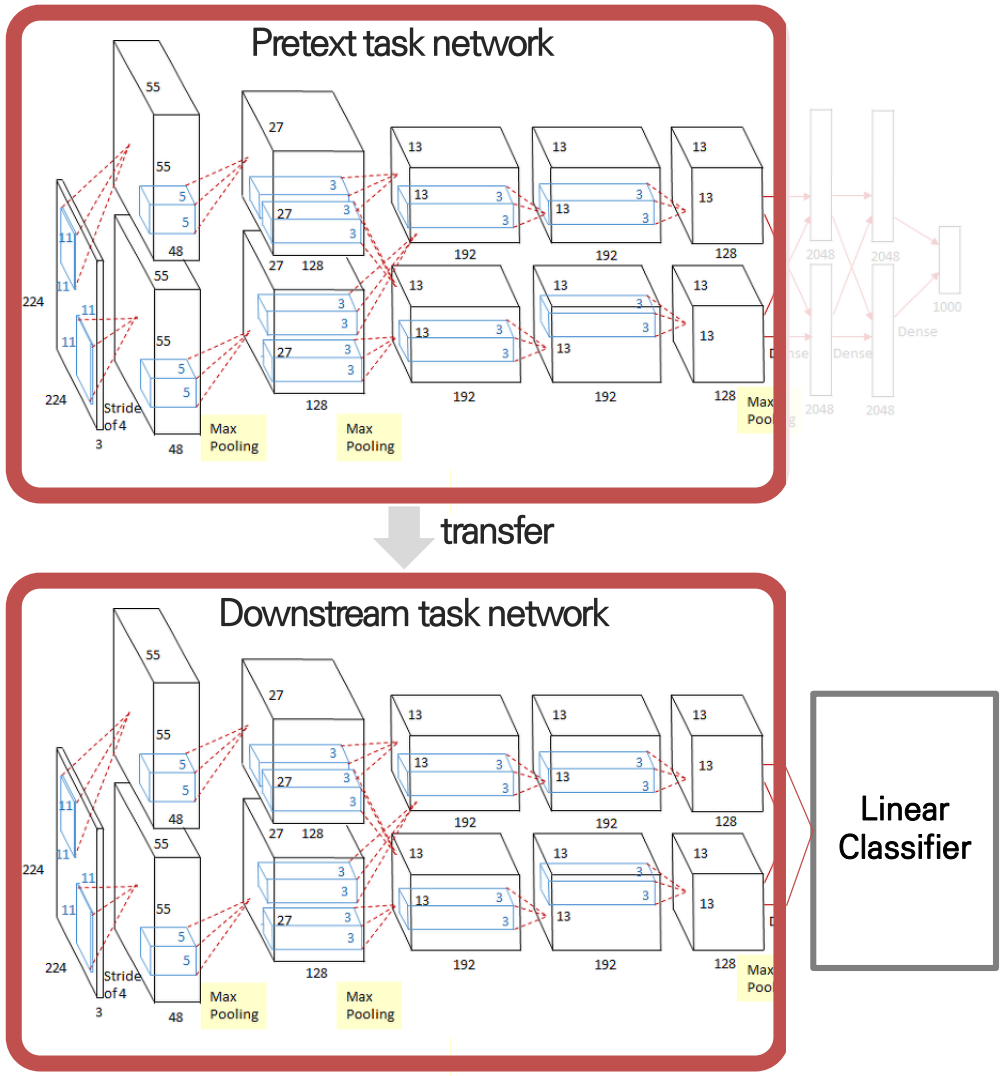
Downstream task evaluation

Transfer learning

〈theoretical framework〉



〈 actual framework 〉



Pre-training 된 부분을 transfer 한 후에 weight들을 freeze하고 linear classifier를 통해 성능 분석

기존의 알려진 label dataset 으로 성능을 평가함 (e.g. ImageNet)

Downstream task evaluation

Transfer learning

❖ Classification accuracy

RotNet evaluation table

Method	Accuracy
Supervised NIN	92.80
Random Init. + conv	72.50
(Ours) RotNet + non-linear	89.06
(Ours) RotNet + conv	**91.16**
(Ours) RotNet + non-linear (fine-tuned)	91.73
(Ours) RotNet + conv (fine-tuned)	92.17
Roto-Scat + SVM Oyallon & Mallat (2015)	82.3
ExemplarCNN Dosovitskiy et al. (2014)	84.3
DCGAN Radford et al. (2015)	82.8
Scattering Oyallon et al. (2017)	84.7

SimCLR evaluation table

Method	Architecture	Param.	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.

Class 개수가 적을 경우 accuracy 사용

Class 개수가 많을 경우 Top-1, Top-5 accuracy 사용

Downstream task evaluation

Transfer learning

❖ Clustering evaluation

Context prediction method

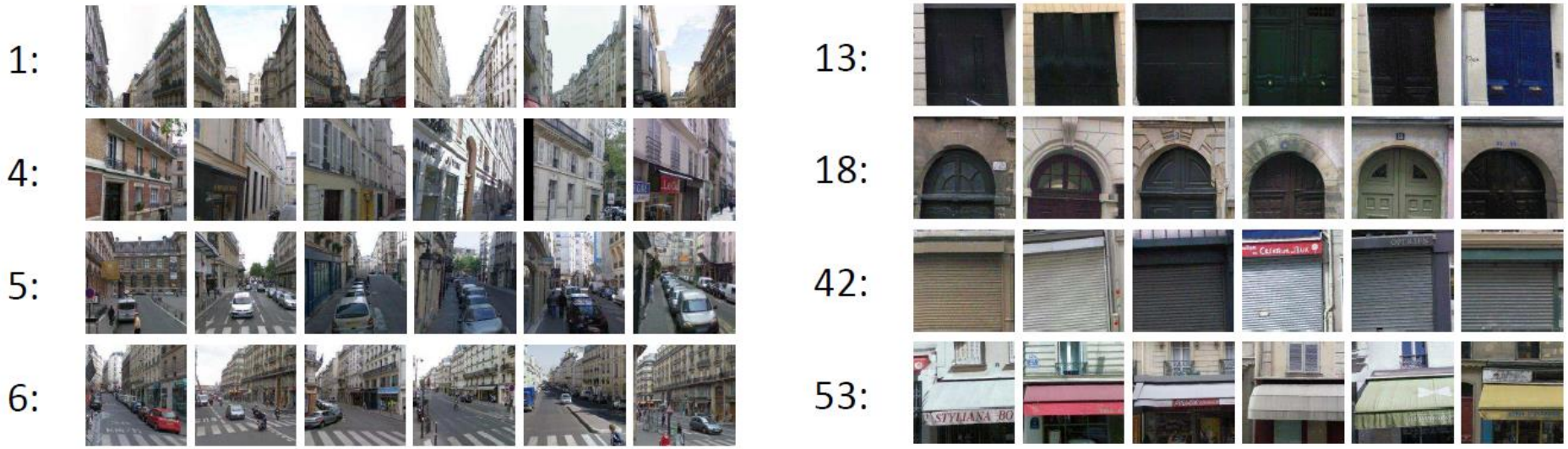


Figure 8. Clusters discovered and automatically ranked via our algorithm (§ 4.4) from the Paris Street View dataset.

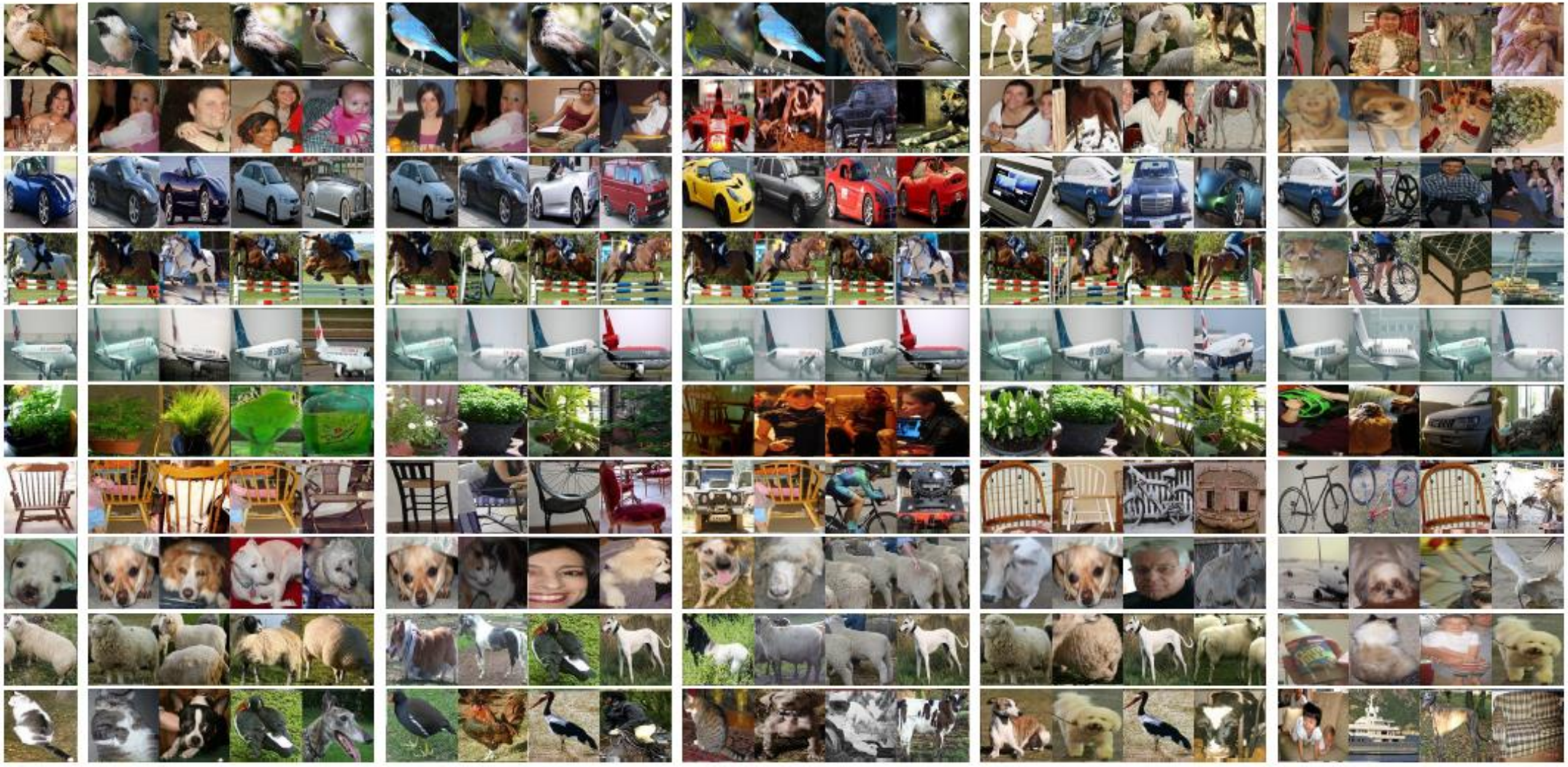


Fig. 5: Image retrieval (qualitative evaluation). (a) query images; (b) top-4 matches with AlexNet; (c) top-4 matches with the CFN trained without blocking chromatic aberration; (d) top-4 matches with Doersch *et al.* [10]; (e) top-4 matches with Wang and Gupta [39]; (f) top-4 matches with AlexNet with random weights.

Jigsaw method

SimCLR

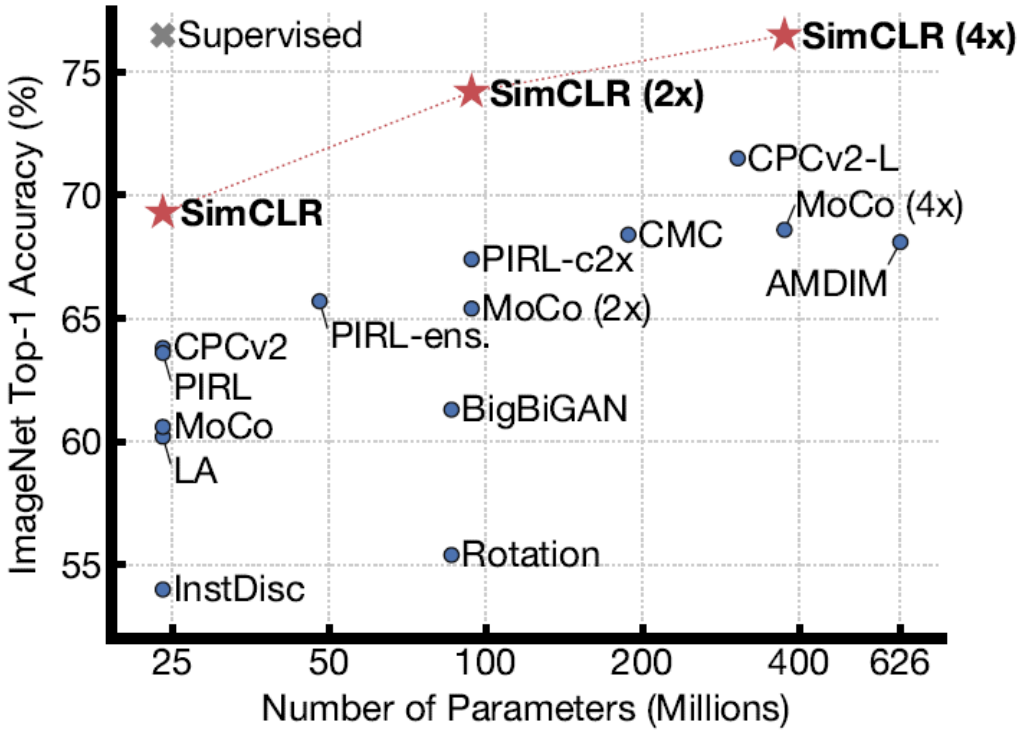
Paper explanation

❖ Simple Framework for Contrastive Learning of Visual Representations(SimCLR), 2020 arXiv

A Simple Framework for Contrastive Learning of Visual Representations

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

- 최근에 발표된 self-supervised representation learning 방법론 중 가장 성능이 우수 (2020년 2월 13일 기준)
- 인공지능 분야의 세계적 석학인 Geoffrey Hinton(토론토 대학교 교수)이 논문에 참여
- Contrastive learning 방법을 적용하였고, architecture를 간단하게 구현



SimCLR

Paper explanation

❖ Method(algorithm)

- The Contrastive Learning Framework

I. Stochastic data augmentation을 진행

1) Random cropping and resize to original
2) Random color distortion
3) Random Gaussian blur

II. Neural network base encoder f(·)를 통과

III. Neural network projection head g(·)로 매핑(mapping)

IV. Contrastive loss function 계산

Algorithm 1 SimCLR's main learning algorithm.

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

SimCLR

Paper explanation

❖ Method(algorithm)

- The Contrastive Learning Framework

I. Stochastic data augmentation을 진행

1) Random cropping and resize to original
2) Random color distortion
3) Random Gaussian blur

II. Neural network base encoder f(·)를 통과

III. Neural network projection head g(·)로 매핑(mapping)

IV. Contrastive loss function 계산

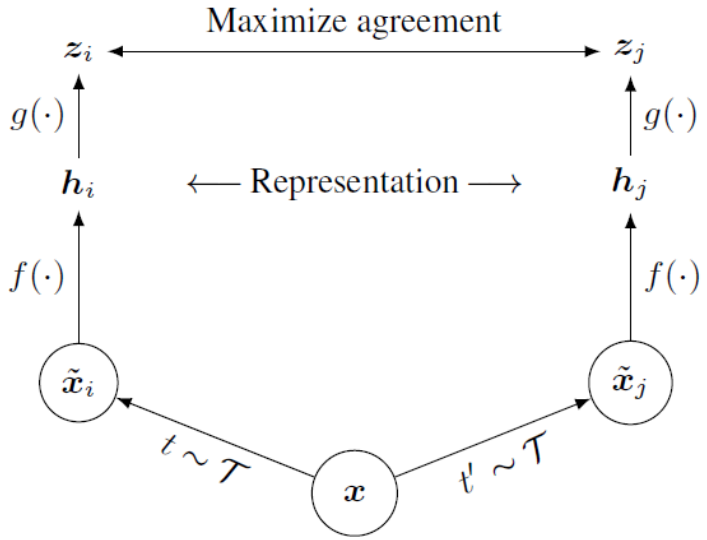


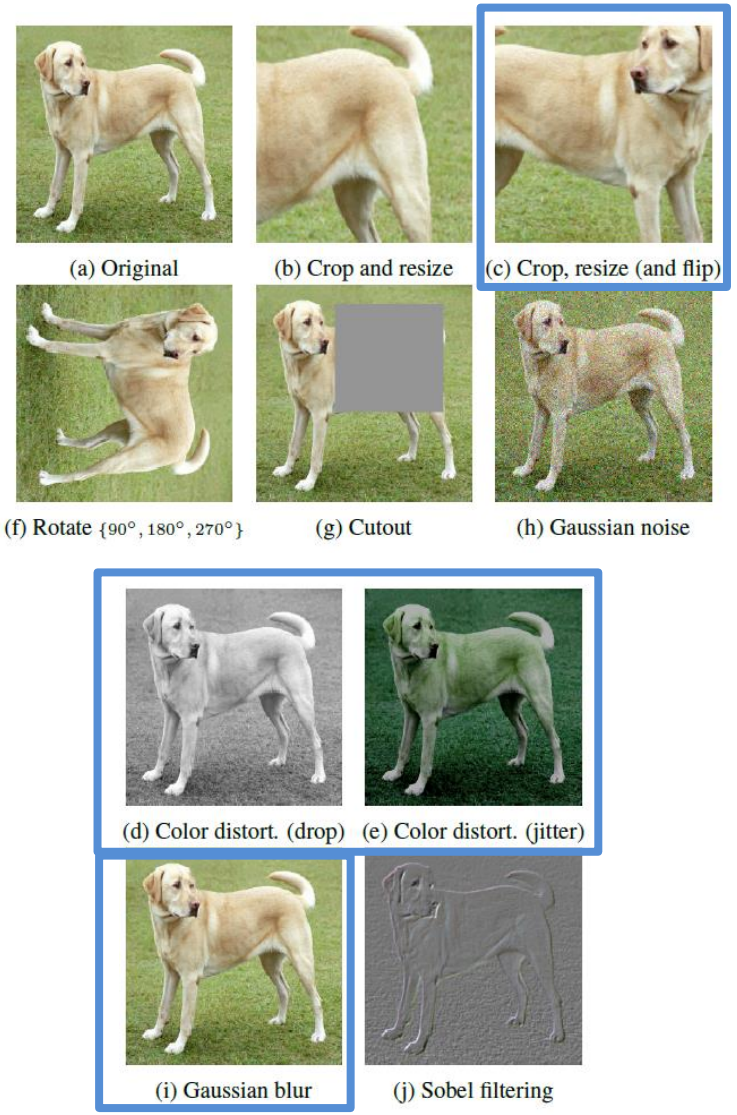
Figure 2. A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we through away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tasks.

SimCLR

Paper explanation

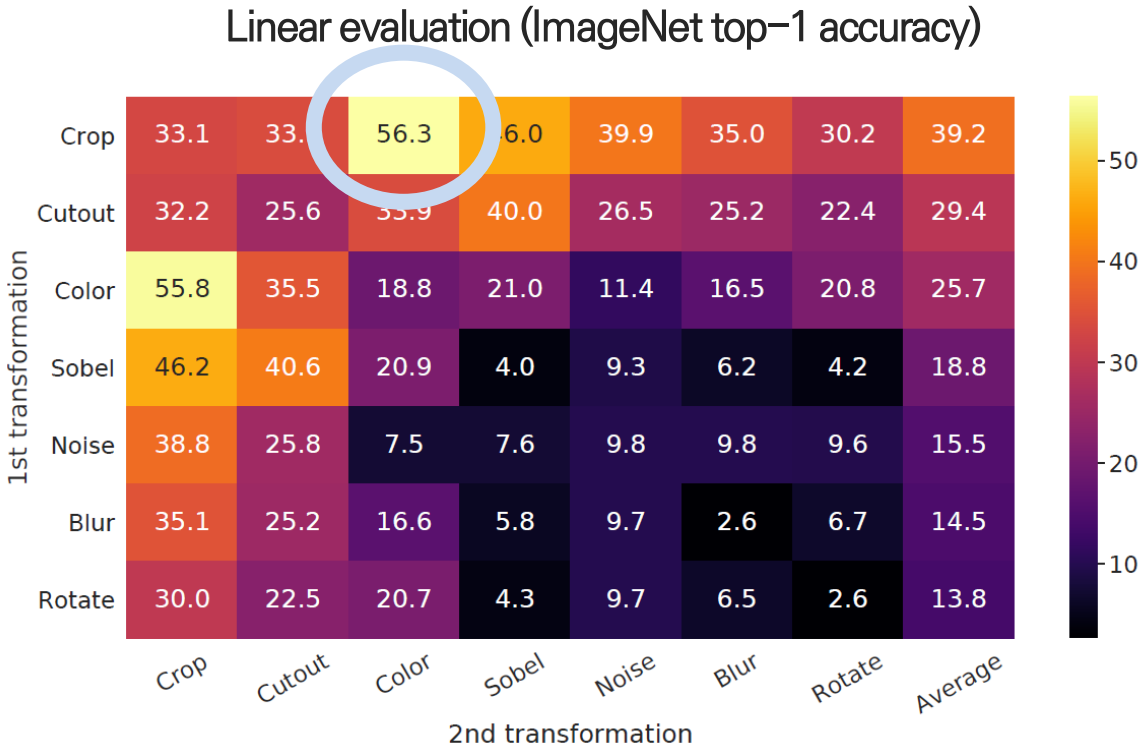
❖ Method(algorithm)

- The Contrastive Learning Framework
  - I. Stochastic data augmentation을 진행
    - 1) Random cropping and resize to original
    - 2) Random color distortion
    - 3) Random Gaussian blur
  - II. Neural network base encoder f(·)를 통과
  - III. Neural network projection head g(·)로 매핑(mapping)
  - IV. Contrastive loss function 계산

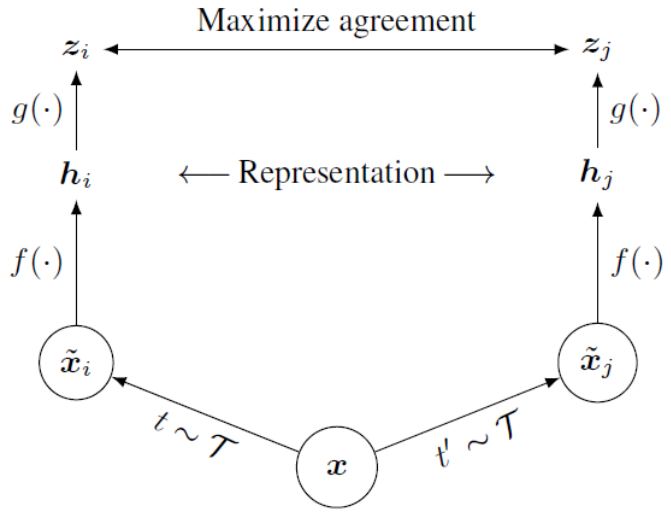


SimCLR

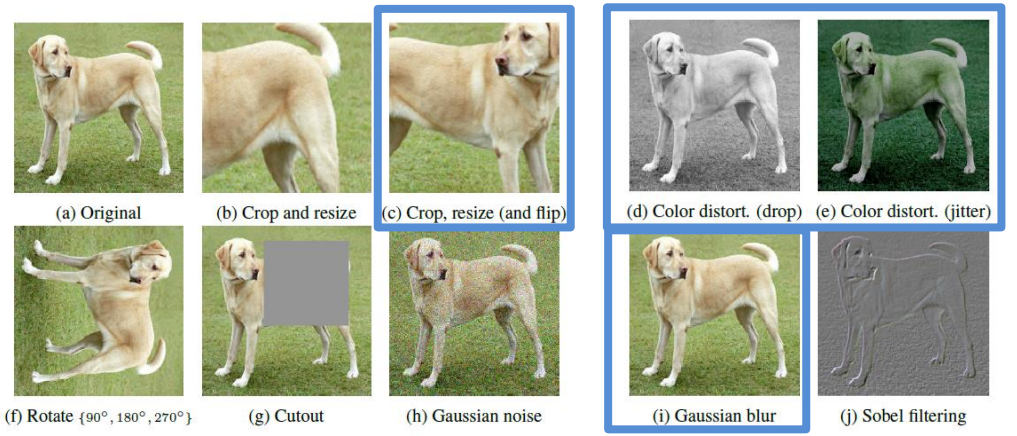
Paper explanation



Crop과 Color의 조합이 가장 우수한 성능을 보여줌



Input x를 augmentation한 결과가 $\tilde{x}_i, \tilde{x}_j$ 로 나타남



SimCLR

Paper explanation

❖ Method(algorithm)

- The Contrastive Learning Framework

I. Stochastic data augmentation을 진행

1) Random cropping and resize to original

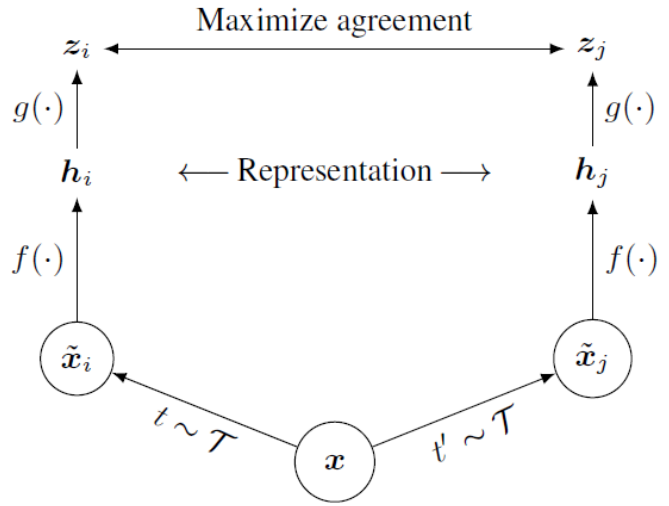
2) Random color distortion

3) Random Gaussian blur

II. Neural network base encoder f(·)를 통과

III. Neural network projection head g(·)로 매핑(mapping)

IV. Contrastive loss function 계산



ResNet-50

SimCLR

Paper explanation

❖ Method(algorithm)

- The Contrastive Learning Framework

I. Stochastic data augmentation을 진행

1) Random cropping and resize to original

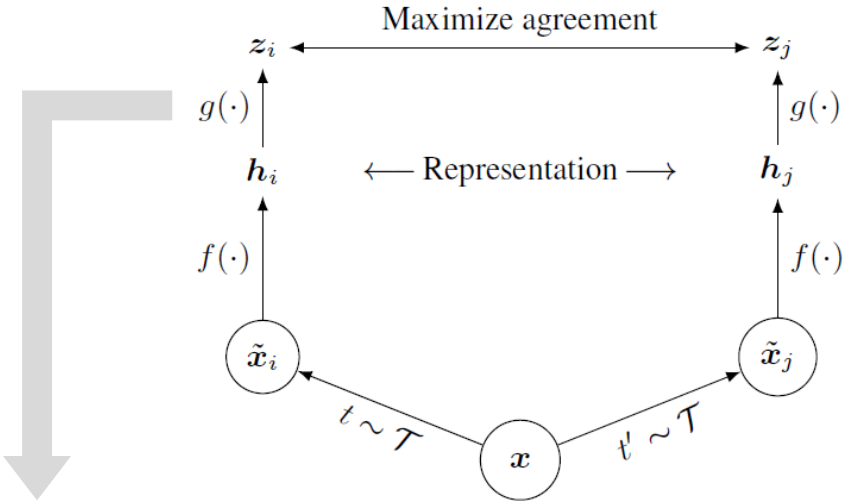
2) Random color distortion

3) Random Gaussian blur

II. Neural network base encoder f(·)를 통과

III. Neural network projection head g(·)로 매핑(mapping)

IV. Contrastive loss function 계산



$$g(h) = W^{(2)}\sigma(W^{(1)}h)$$

SimCLR

Paper explanation

❖ Method(algorithm)

- The Contrastive Learning Framework

I. Stochastic data augmentation을 진행

1) Random cropping and resize to original

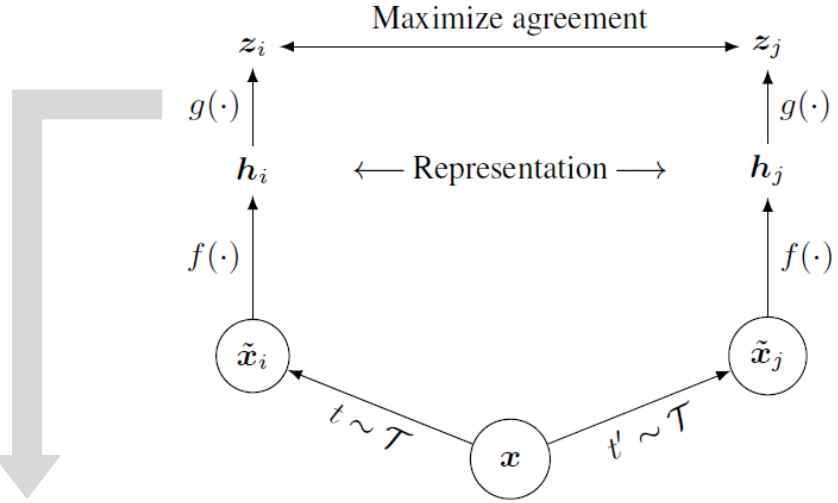
2) Random color distortion

3) Random Gaussian blur

II. Neural network base encoder f(·)를 통과

III. Neural network projection head g(·)로 매핑(mapping)

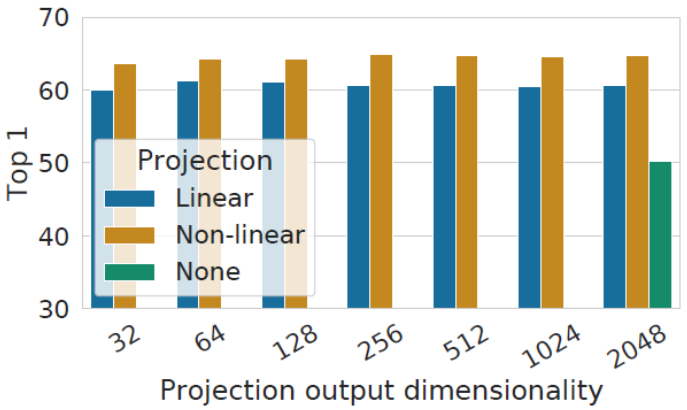
IV. Contrastive loss function 계산



$$g(h) = W^{(2)}\sigma(W^{(1)}h)$$

Contrastive loss를 계산하는 space로 mapping하는 함수

Linear말고 nonlinear함수를 쓴 이유는 실험적으로 성능이 더 좋았기 때문



SimCLR

Paper explanation

❖ Method(algorithm)

- The Contrastive Learning Framework

I. Stochastic data augmentation을 진행

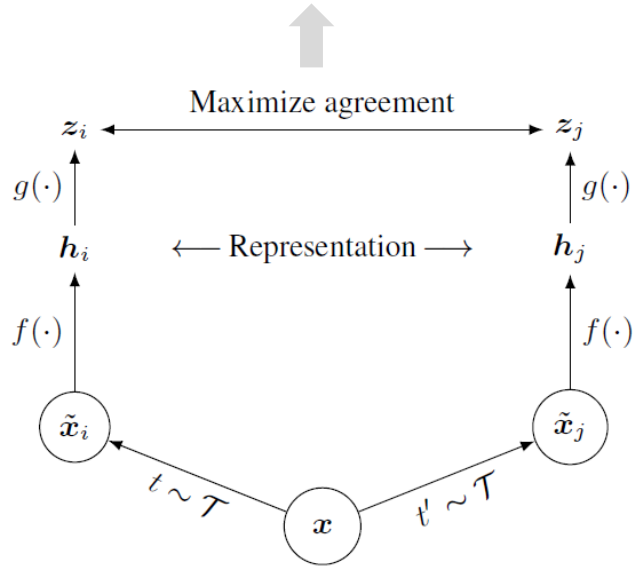
1) Random cropping and resize to original
2) Random color distortion
3) Random Gaussian blur

II. Neural network base encoder f(·)를 통과

III. Neural network projection head g(·)로 매핑(mapping)

IV. Contrastive loss function 계산

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



SimCLR

Paper explanation

같은 image에서 나온 z_i, z_j의 유사도(cosine similarity를 이용)는 클수록 좋고, 다른 image에서 나온 z_i, z_k의 유사도는 작을수록 좋음

또한 negative sampling을 통해 batch 단위로 학습하여 모든 z 간의 유사도를 구하지 않아도 됨

❖ Method(algorithm)

- The Contrastive Learning Framework

I. Stochastic data augmentation을 진행

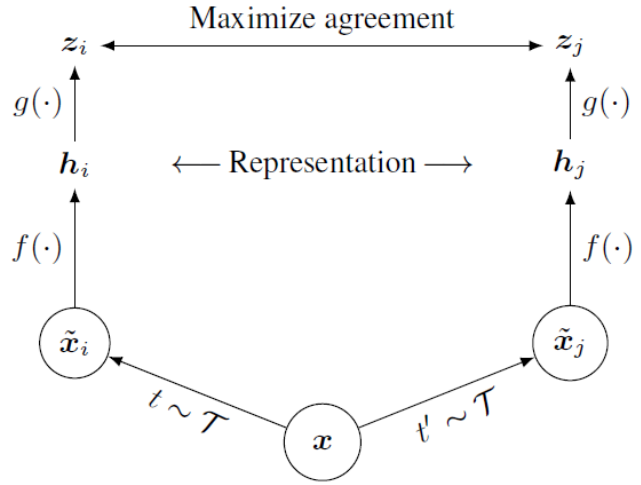
1) Random cropping and resize to original
2) Random color distortion
3) Random Gaussian blur

II. Neural network base encoder f(·)를 통과

III. Neural network projection head g(·)로 매핑(mapping)

IV. Contrastive loss function 계산

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



SimCLR

Paper explanation

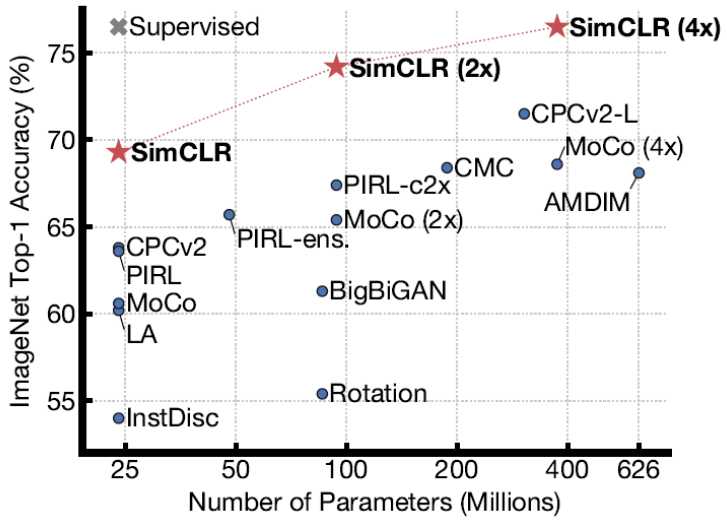
❖ Results

- Label data를 일부 사용하여 모델링하는 semi-supervised learning 방식의 State-of-the-art(SOTA) 성능을 뛰어넘었음

- Self-supervised learning 방법 중에서도 가장 우수한 성능을 보임

Method	Architecture	Label fraction 1%	Label fraction 10%
		Top 5	
Methods using other label-propagation:			
Pseudo-label	ResNet50	51.6	82.4
VAT+Entropy Min.	ResNet50	47.0	83.4
UDA (w. RandAug)	ResNet50	-	88.5
FixMatch (w. RandAug)	ResNet50	-	89.1
S4L (Rot+VAT+En. M.)	ResNet50 (4×)	-	91.2
Methods using representation learning only:			
InstDisc	ResNet50	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.



Method	Architecture	Param.	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.

SimCLR

Paper explanation

❖ Results

- ImageNet으로 pre-train한 ResNet-50(4x)를 이용한 실험 결과
- 거의 대부분의 dataset에서 훌륭한 성능을 보여주고 있음

	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

Table 8. Comparison of transfer learning performance of our self-supervised approach with supervised baselines across 12 natural image classification datasets, for ResNet-50 $(4\times)$ models pretrained on ImageNet. Results not significantly worse than the best ($p > 0.05$, permutation test) are shown in bold. See Appendix B.6 for experimental details and results with standard ResNet-50.

Conclusion

❖ Comments

- 양질의 dataset을 만드는 것이 현실적으로 어려운데, 최근 unsupervised learning 방법 중 하나인 self-supervised learning 방식의 연구가 상당부분 진행중임을 확인

- 불과 몇 년 사이에 기초 연구 단계에서 SOTA 성능을 보이는 SimCLR까지 나온 것을 보아 딥러닝 분야의 발전속도를 다시 한번 체감하게 되었음

- 현업에서도 잘 정리되지 않은 unlabeled dataset이 상당히 많은 것으로 알고 있는데, 최신 논문들을 적용하여 supervised learning 못지않은 성능을 기대해볼 수도 있을 것이라 생각함

- 어떻게 보면 input data로 input data를 설명한다는 것이 가장 합리적인 방법이라는 생각이 들었음. Label은 사실상 누군가의 편향이 담길 수 있다고 생각

Thank you