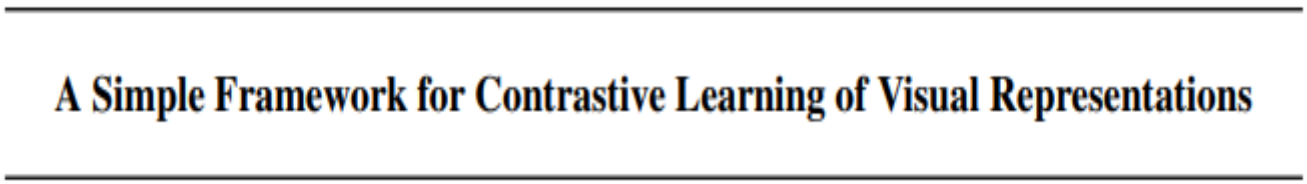
# A Simple Framework for Contrastive Learning of Visual Representations

2023. 05. 26

# SimCLR (2020, ICML)

❖ A Simple Framework for Contrastive Learning of Visual Representations

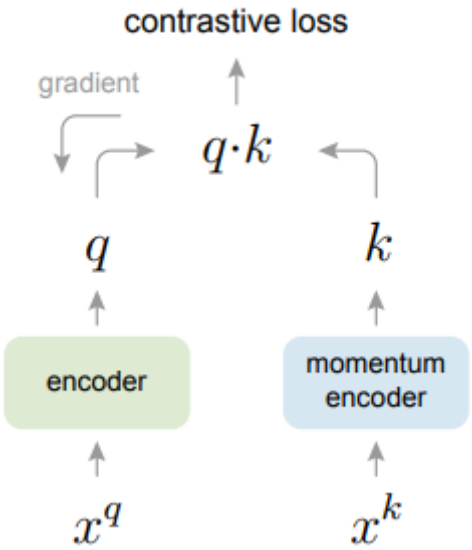
- 2020년에 ICML에 게재되었으며, 2023년 05월 26일 기준 10119회 인용됨
- 많은 실험을 통해 SimCLR의 우수성을 보여주는 것이 본 논문에서의 특이점
- 간단한 프레임워크를 사용해 기존 대조 학습 방법론 대비 우수한 성능을 보임
- 대조 학습을 사용해 유용한 표현을 학습할 수 있는 요인을 이해하기 위해 크게 세 가지 연구 진행
  1. 데이터 증강의 구성은 효과적인 예측 작업에 중요한 역할을 하는 것을 확인하는 연구
  2. 비선형 변환을 도입하여 학습된 표현의 품질을 크게 향상하는 연구
  3. 더 큰 배치 사이즈와 더 많은 훈련 단계에서 지도 학습보다 더 많은 benefit이 있는 것을 확인하는 연구



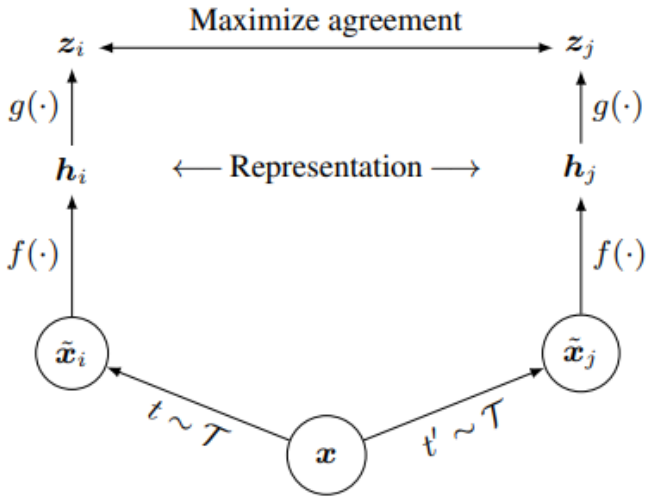
# Background

❖ MoCo(2020,CVPR) vs SimCLR(2020,ICML) 차이점

- MoCo: memory bank를 사용하고 target network를 업데이트할 때 momentum update (EMA)를 사용
- SimCLR: memory bank를 사용하지 않고 동일한 encoder를 사용함




MoCo

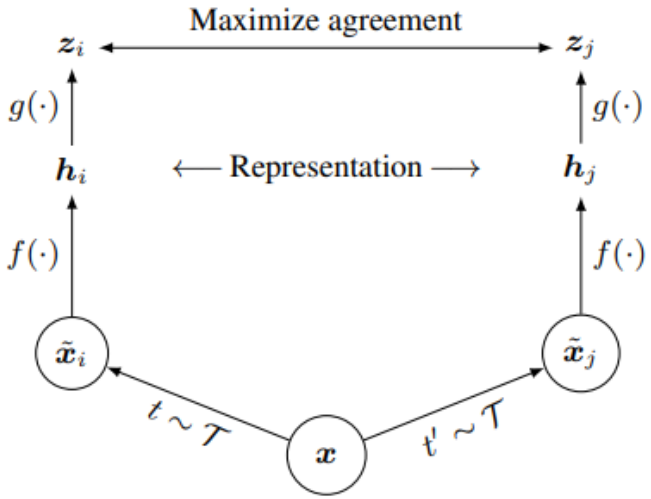



SimCLR

# Proposed Method

## ❖ Overall Architecture

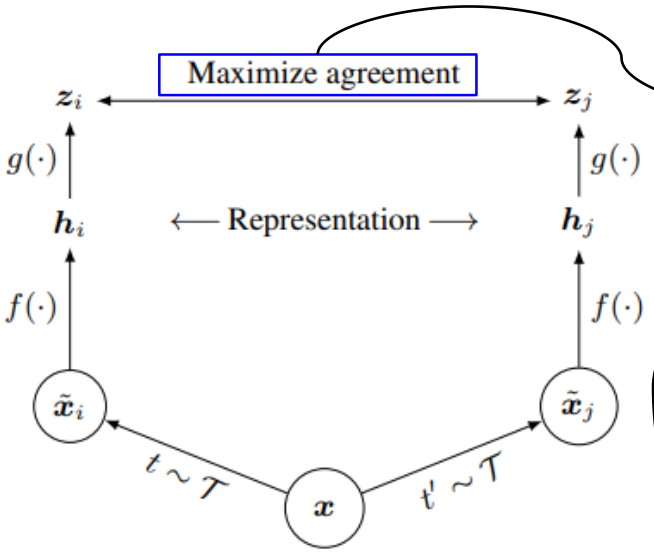
- 동일한 encoder($f$), projection network($g$)를 사용
- minibatch 내에서 뽑은 데이터($x$)에 서로 다른 데이터 증강 기법을 적용한 데이터($\tilde{x}_i, \tilde{x}_j$) 외에 나머지를 negative samples로 간주함




SimCLR

# Proposed Method

❖ Overall Architecture – loss function




SimCLR

Cosine similarity

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

$$sim(u, v) = u^T v / \|u\| \|v\|$$

Indicator function,만약, k가 i와 다르다면 1, 같다면 0

# Proposed Method

## ❖ Overall Architecture – pseudo code

**Algorithm 1** SimCLR's main learning algorithm.

**input:** batch size $N$, constant $\tau$, structure of $f, g, \mathcal{T}$.
**for** sampled minibatch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  **for all** $k \in \{1, \ldots, N\}$ ~~**do**~~
    draw two augmentation functions $t \sim \mathcal{T}, t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # 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 $g(\cdot)$

we sequentially apply three simple augmentations: random cropping followed by resize back to the original size, random color distortions, and random Gaussian blur
→ 본 논문에서 확인할 수 있듯이 서로 다른 데이터 증강 기법을 적용할 때, sequential하게 3개의 데이터 증강 기법 적용

Batch size

동일한 encoder와 projection head를 사용한 것을 확인
논문에서는, We opt for simplicity and adopt the commonly used ResNet, 이러한 이유로 Resnet을 사용했고 projection head로는 히든 레이어를 한 개 가진 MLP(with using ReLu)를 사용함

서로 다른 데이터 증강 기법 적용: N→ 2N

Positive pair 끼리 가까워지도록

Positive와 Negative sample 끼리 멀어지도록
→ 즉 분모의 의미는 i가 만약 k와 다르다면, 이 때 k가 바로 negative sample을 의미한다. i가 positive sample을 의미하기 때문에

2k와 2k−1일 때의 의미는 서로 다른 데이터 증강 기법을 적용한 positive pair를 의미함. 이 때 2k와 나머지(negative pair)간에는 멀어지도록, 2k−1과 나머지(negative pair)와 멀어지도록 하기 위해 $l(2k-1,2k)$&$l(2k,2k-1)$을 각각 진행함. 결국 해당 $L$를 최소화하는 방향으로 진행됨 → 분모는 작아지게(positive와 negative 멀어지게) 분자는 커지도록(positive 끼리 가까워지도록) 학습 진행

# Experiment

❖ Experiment 1: color distortion의 strength에 따른 SimCLR와 지도 학습(ResNet-50) 성능 비교

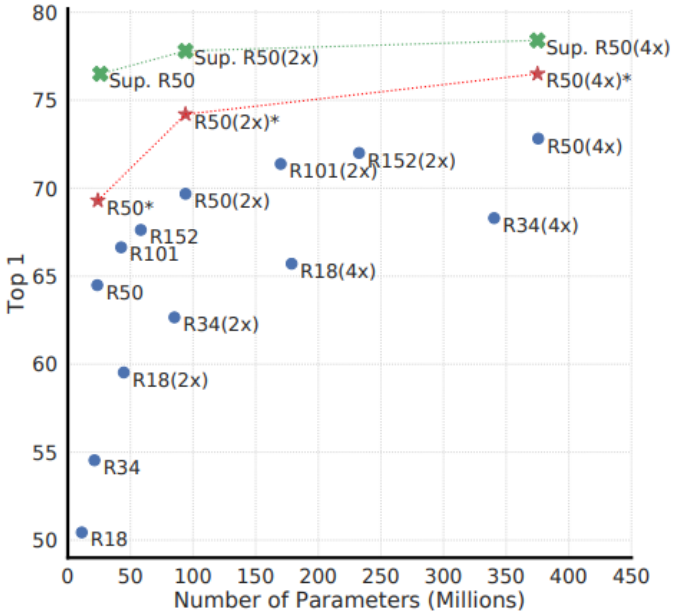
- 본 실험의 목적은 color augmentation의 중요성을 설명하기 위함
- 단순 분류 task(linear evaluation)에서 color distortion의 정도 강해짐에 따라 SimCLR의 성능 향상
- 반면에, 지도 학습은 color distortion이 강해짐에 따라 성능이 개선되 지도 떨어지 지도 않음
- 결론: 비지도 대조 학습이 지도 학습보다 Stronger (color) data augmentation을 사용했을 때 더 많은 benefit 이 있음
- 참고 1) SimCLR의 backbone은 ResNet-50
- 참고 2) Auto Augment는 지도 학습을 사용해 찾은 정교한 데이터 증강 기법인데, 비지도 대조 학습에서 간단한 cropping + (stronger) color distortion에 비해 성능이 좋지 않음

| Methods | Color distortion strength | | | | | AutoAug |
|---|---|---|---|---|---|---|
| | 1/8 | 1/4 | 1/2 | 1 | 1 (+Blur) | |
| 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 |

# Experiment

## ❖ Experiment 2: backbone 모델이 더 커질 수록 비지도 대조 학습의 성능이 우수해짐

- 본 실험의 목적은 모델이 커질 수록 비지도 대조 학습에 끼치는 영향을 확인하기 위함
- 간단한 분류 task에서 모델의 depth와 width가 커질 수록 성능이 향상하고 지도 학습과 거의 유사한 성능을 보임
- 결론: depth와 width가 커질 수록 성능 향상됨
- 참고 ) 초록색 점은 90 epoch동안 학습된 지도 학습 ResNET / 빨간색 선은 1000 epoch 동안 학습된 SimCLR / 파란색 점은 100 epoch 동안 학습된 SimCLR

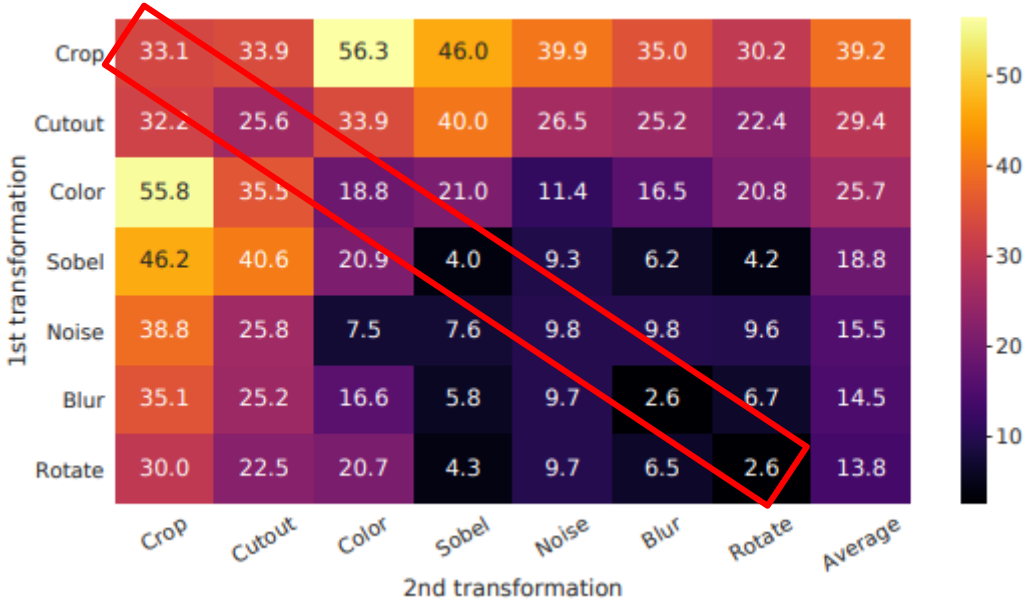



Data Mining
Quality Analytics

# Experiment

❖ Experiment 3: 데이터 증강 기법의 구성에 따른 성능 확인 (ImageNet top-1 accuracy)

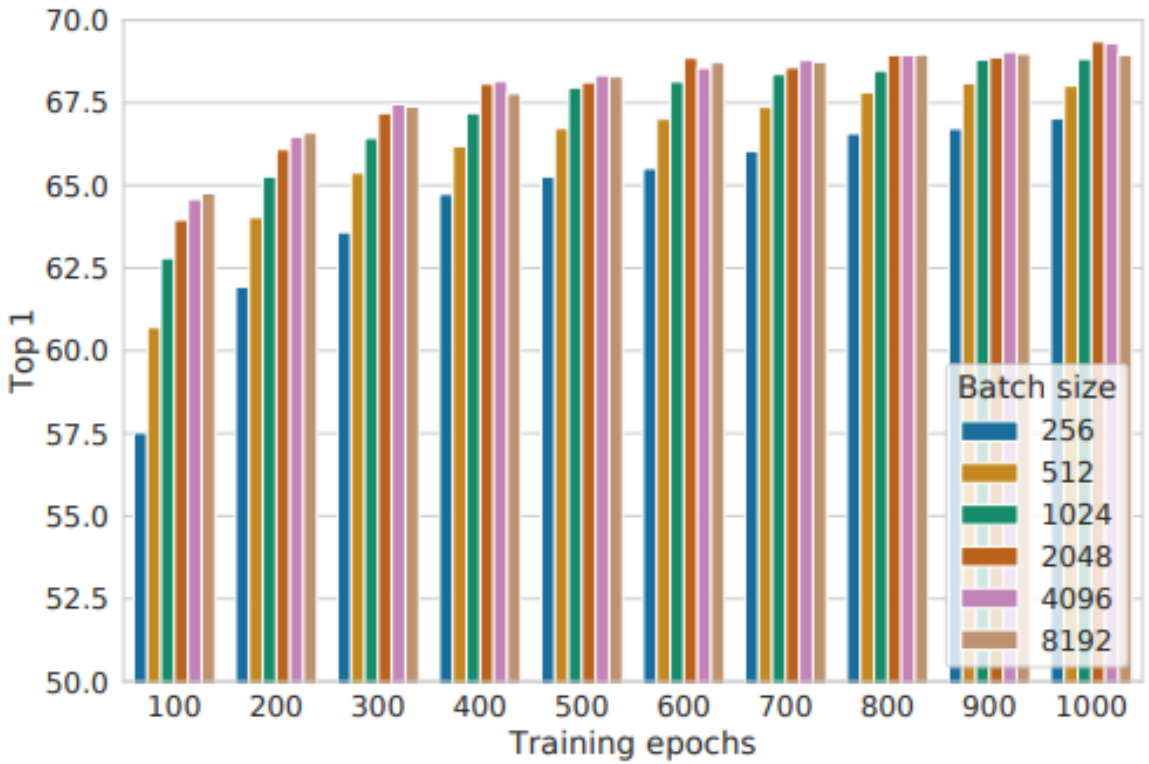
- 본 실험의 목적은 데이터 증강 기법이 단일로 사용되었을 때, 두 개의 구성으로 이루어졌을 때 성능 확인
- 대각선이 단일로 데이터 증강 기법을 적용하였을 때 성능
- 대각선을 제외한 나머지는 두 데이터 증강 기법을 함께 적용했을 때 성능 / 마지막 열은 각 행의 평균
- 결론: 단일 변환을 사용해서는 좋은 표현을 학습하기에 충분하지 않지만 두 개 데이터 증강 기법을 함께 적용했을 때 task가 더 어려워졌지만 더 질 좋은 표현을 학습함



# Experiment

❖ Experiment 4: 더 큰 batch size와 더 오랜 학습 시간에 따른 성능 확인

- 결론: batch size가 커질 수록 학습 시간이 길어질 수록 더 좋은 성능을 보임



# Experiment

❖ Experiment 5 & 6: ImageNet 데이터 셋을 사용해 다른 방법과 성능 비교

- "SimCLR is better than others"

다른 자기 지도학습 방법과 비교

| 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** |

Label의 사용을 조절해 자기 지도학습 및 준지도 학습과 성능 비교

| Method | Architecture | Label fraction (Top 5) 1% | 10% |
|---|---|---|---|
| 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** |

# Experiment

❖ Experiment 7: 여러 데이터 셋을 활용해 SimCLR와 지도 학습 간의 성능 비교

| | 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 |