dive in contrastive learning

data augmentation and vsiual representations

outline

- A Simple Framework for Contrastive Learning of Visual Representations
- Supervised Contrastive Learning
- Weakly Supervised Contrastive Learning
- Less can be more in contrastive learning
  - https://openreview.net/pdf?id=U2exBrf_SJh

SimCLR

Contrastive Learning of Visual Representations
(Google Research)

Abstract

- 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-ofthe-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.

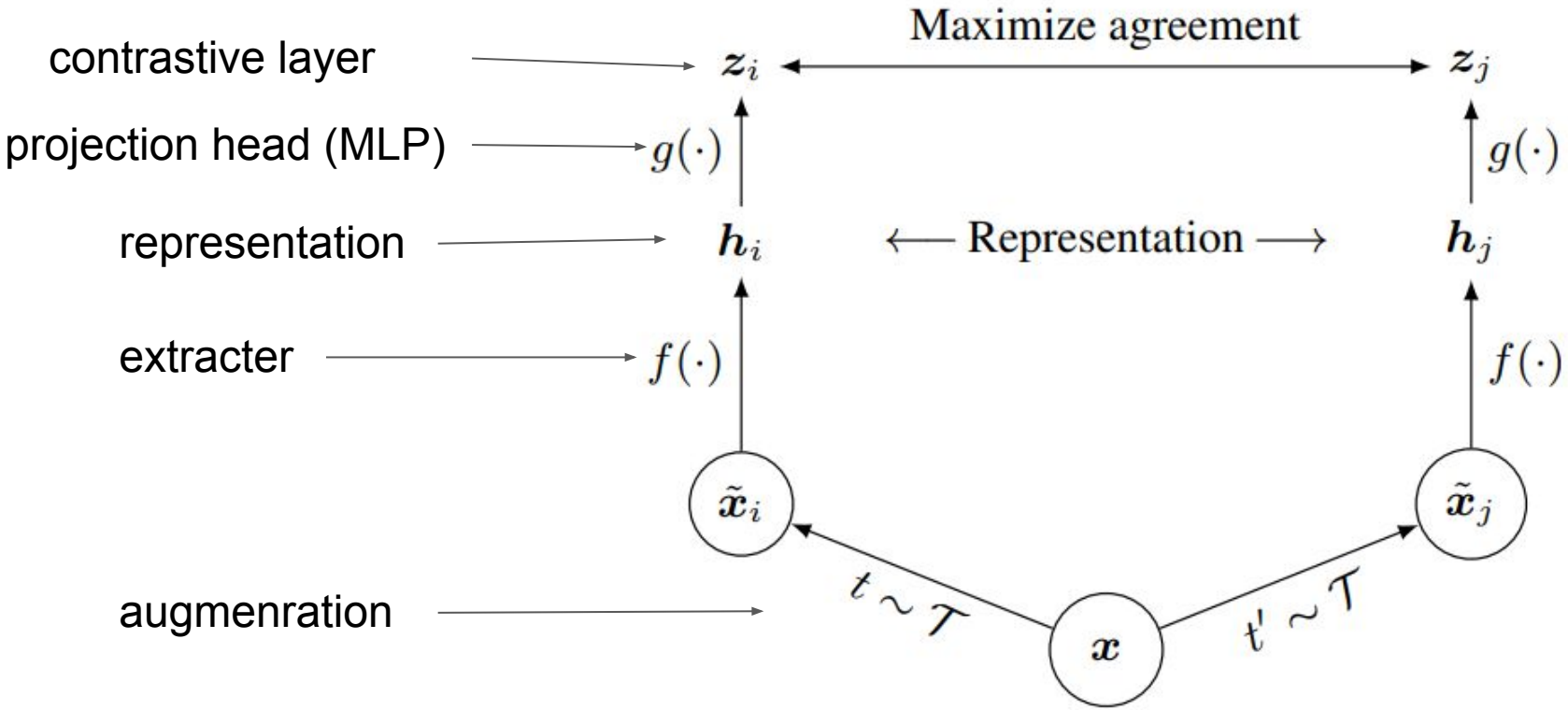
major components

- composition of data augmentations plays a critical role in defining effective predictive tasks.
- introducing a learnable nonlinear transformation between the representation and the contrastive loss substantially improves the quality of the learned representations.
- Representation learning with contrastive cross entropy loss benefits from normalized embeddings and an appropriately adjusted temperature parameter.
- contrastive learning benefits from larger batch sizes and more training steps compared to supervised learning

The Contrastive Learning Framework

- SimCLR learns representations by maximizing agreement between differently augmented views of the same data example via a contrastive loss in the latent space.

framework


contrastive layer
projection head (MLP)
representation
extracter
augmenration
Maximize agreement
z_i
z_j
$g(\cdot)$
$g(\cdot)$
h_i
Representation
h_j
$f(\cdot)$
$f(\cdot)$
$\tilde{x}_i$
$\tilde{x}_j$
$t \sim \mathcal{T}$
$t' \sim \mathcal{T}$
x

data augmentation

- A stochastic data augmentation module that transforms any given data example randomly resulting in two correlated views of the same example, which we consider as a positive pair.
- sequentially apply three simple augmentations: random cropping followed by resize back to the original size, random color distortions, and random Gaussian blur.

extracter

- A neural network base encoder $f(\cdot)$ that extracts representation vectors from augmented data examples. Our framework allows various choices of the network architecture without any constraints.

projection head

- A small neural network projection head g(·) that maps representations to the space where contrastive loss is applied. We use a MLP with one hidden layer to obtain

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

where σ is a ReLU nonlinearity. we find it beneficial to define the contrastive loss on z rather than h.

contrastive loss

A *contrastive loss function* defined for a contrastive prediction task. Given a set $\{\tilde{\boldsymbol{x}}_k\}$ including a positive pair of examples $\tilde{\boldsymbol{x}}_i$ and $\tilde{\boldsymbol{x}}_j$, the *contrastive prediction task* aims to identify $\tilde{\boldsymbol{x}}_j$ in $\{\tilde{\boldsymbol{x}}_k\}_{k \neq i}$ for a given $\tilde{\boldsymbol{x}}_i$.

contrastive structure

- We randomly sample a minibatch of N examples and define the contrastive prediction task on pairs of augmented examples derived from the minibatch, resulting in 2N data points. We do not sample negative examples explicitly. Instead, given a positive pair, we treat the other 2(N − 1) augmented examples within a minibatch as negative examples.

loss function (NT-Xent)

u and v are normalized

(i,j) denoted positive pair

$$\text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$$

cos sim

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

Training with Large Batch Size

- we vary the training batch size N from 256 to 8192
- Training with large batch size may be unstable when using standard SGD/Momentum with linear learning rate scaling. To stabilize the training, we use the LARS optimizer for all batch sizes.

Default setting

- Unless otherwise specified, for data augmentation we use random crop and resize (with random flip), color distortions, and Gaussian blur.
- 2-layer MLP projection head to project the representation to a 128-dimensional latent space.
- As the loss, we use NT-Xent.

Data Augmentation for CLR

(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

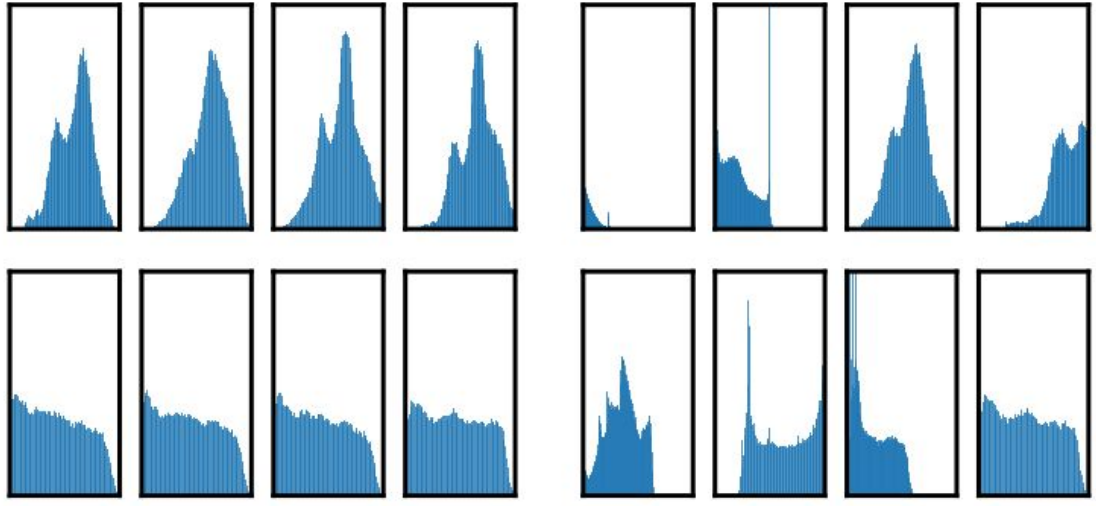
Representation Learning (related)

- Many existing approaches define contrastive prediction tasks by changing the architecture.
- neighboring view prediction via a fixed image splitting procedure and a context aggregation network.
- We show that this complexity can be avoided by performing simple random cropping (with resizing) of target images

data augmentation operation

- only to one branch of the framework in Figure 2, while leaving the other branch as the identity (i.e. t(x) = x). this asymmetric data augmentation hurts the performance.
- When composing augmentations, the contrastive prediction task becomes harder, but the quality of representation improves dramatically.
- it is critical to compose cropping with color distortion in order to learn generalizable features.

reason



(a) Without color distortion. (b) With color distortion.

Figure 6. Histograms of pixel intensities (over all channels) for different crops of two different images (i.e. two rows). The image for the first row is from Figure 4. All axes have the same range.

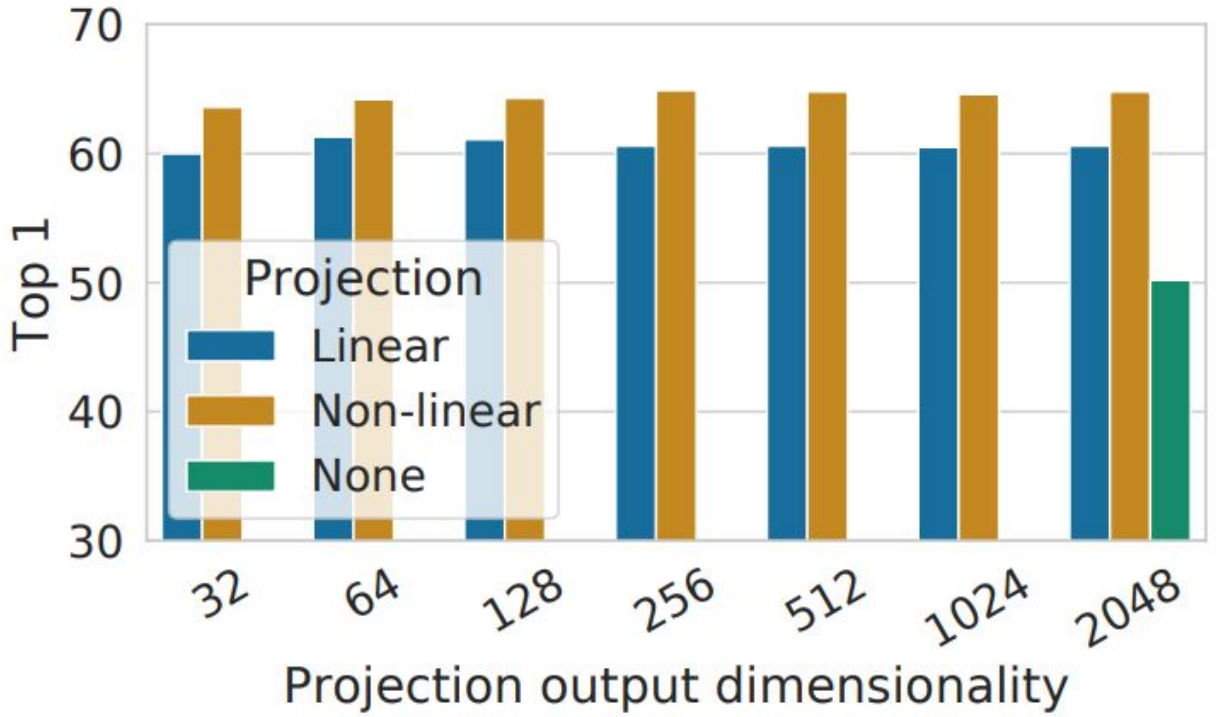
data augmentation level

- Stronger color augmentation substantially improves the linear evaluation of the learned unsupervised models.
- When training supervised models with the same set of augmentations, we observe that stronger color augmentation does not improve or even hurts their performance.
- unsupervised contrastive learning benefits from stronger (color) data augmentation than supervised learning.

nonlinear projection head

- identity mapping
- linear projection
- nonlinear projection (with one additional hidden layer and ReLU)
- We observe that a nonlinear projection is better than a linear projection (+3%), and much better than no projection (>10%).
- When a projection head is used, similar results are observed regardless of output dimension.

projection compare



hypothesis

- g can remove information that may be useful for the downstream task, such as the color or orientation of objects.
- By leveraging the nonlinear transformation g(·), more information can be formed and maintained in h.

What to predict?	Random guess	Representation $\boldsymbol{h}$	Representation $g(\boldsymbol{h})$
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

Table 3. Accuracy of training additional MLPs on different representations to predict the transformation applied. Other than crop

Supervised Contrastive Learning

Weakly Supervised Contrastive Learning (ICCV)

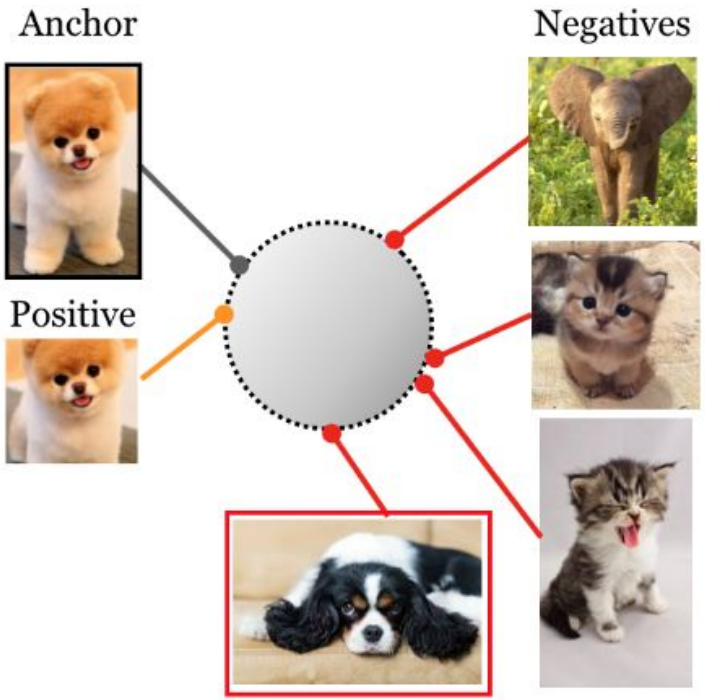
SupCon

Supervised Contrastive Learning (ICCV)

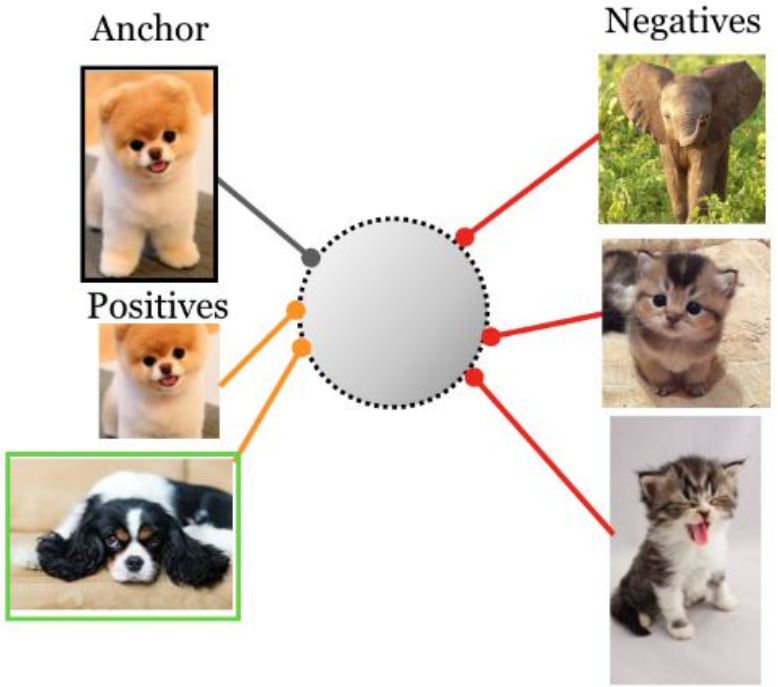
Introduction

- A number of works have explored shortcomings of this loss, such as lack of robustness to noisy labels and the possibility of poor margins, leading to reduced generalization performance. However, in practice, most proposed alternatives have not worked better for large-scale datasets, such as ImageNet.

example



Self Supervised Contrastive



Supervised Contrastive

contrastive loss in supervised learning

- These positives are drawn from samples of the same class as the anchor, rather than being data augmentations of the anchor, as done in self-supervised learning.
- The use of many positives and many negatives for each anchor allows us to achieve state of the art performance without the need for hard negative mining, which can be difficult to tune properly

Method

- Given an input batch of data, we first apply data augmentation twice to obtain two copies of the batch. Both copies are forward propagated through the encoder network to obtain a 2048-dimensional normalized embedding.
- During training, this representation is further propagated through a projection network that is discarded at inference time.

Representation Learning Framework

- Data Augmentation module $Aug(\cdot)$
  - For each input sample, x, we generate two random augmentations, $\widetilde{x} = Aug(x)$, each of which represents a different view of the data and contains some subset of the information in the original sample.

Representation Learning Framework

- Encoder Network $Enc(\cdot)$
  - which maps x to a representation vector, $r = Enc(x) \in R^{\mathrm{D_E}}$. Both augmented samples are separately input to the same encoder, resulting in a pair of representation vectors. r is normalized to the unit hypersphere in $R^{\mathrm{D_E}}$. Consistent with the findings of, our analysis and experiments show that this normalization improves top-1 accuracy.

Representation Learning Framework

- Projection Network $Proj(\cdot)$
  - which maps r to a vector $z = Proj(r) \in R^{\mathrm{D_P}}$. we leave to future work the investigation of optimal $Proj(\cdot)$ architectures. We again normalize the output of this network to lie on the unit hypersphere, which enables using an inner product to measure distances in the projection space.

Self-Supervised Contrastive Loss Functions

$$\boldsymbol{z}_\ell = Proj(Enc(\tilde{\boldsymbol{x}}_\ell)) \in \mathcal{R}^{D_P}$$

$$j(i) \qquad \{k \in A(i) \setminus \{j(i)\}$$

postive example

negtive example

$$\mathcal{L}^{self} = \sum_{i \in I} \mathcal{L}_i^{self} = -\sum_{i \in I} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_{j(i)}/\tau\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a/\tau\right)}$$

Supervised Contrastive Loss Functions

$$\mathcal{L}_{out}^{sup} = \sum_{i \in I} \mathcal{L}_{out,i}^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau\right)}$$

$$\mathcal{L}_{in}^{sup} = \sum_{i \in I} \mathcal{L}_{in,i}^{sup} = \sum_{i \in I} -\log \left\{ \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_p / \tau\right)}{\sum_{a \in A(i)} \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_a / \tau\right)} \right\}$$

Supervised Contrastive Loss

- Generalization to an arbitrary number of positives
  - for any anchor, all positives in a multiviewed batch (i.e., the augmentation-based sample as well as any of the remaining samples with the same label) contribute to the numerator. For randomly-generated batches whose size is large with respect to the number of classes, multiple additional terms will be present (on average, N/C, where C is the number of classes). The supervised losses encourage the encoder to give closely aligned representations to all entries from the same class, resulting in a more robust clustering of the representation space than that generated from Eq. 1

Supervised Contrastive Loss

- contrastive power increases with more negatives
  - Eqs. 2 and 3 both preserve the summation over negatives in the contrastive denominator of Eq. 1. This form is largely motivated by noise contrastive estimation and N-pair losses, wherein the ability to discriminate between signal and noise (negatives) is improved by adding more examples of negatives. This property is important for representation learning via self-supervised contrastive learning, with many papers showing increased performance with increasing number of negatives.

Less can be more

- It has been shown empirically that large batch sizes are needed to achieve good performance, which led the the belief that a large number of negatives is preferable.
- Surprisingly, we discover that for a fixed batch size performance actually degrades as the number of negatives is increased. We also show that using fewer negatives can lead to a better signal-to-noise ratio for the model gradients, which could explain the improved performance.

ref : https://www.deepmind.com/publications/less-can-be-more-in-contrastive-learning

Supervised Contrastive Loss

- Intrinsic ability to perform hard positive/negative mining
  - The gradient contributions from hard positives/negatives (i.e., ones against which continuing to contrast the anchor greatly benefits the encoder) are large while those for easy positives/negatives (i.e., ones against which continuing to contrast the anchor only weakly benefits the encoder) are small. Furthermore, for hard positives, the effect increases (asymptotically) as the number of negatives does
  - Eqs. 2 and 3 both preserve this useful property and generalize it to all positives.

compare

Loss	Top-1
$\mathcal{L}_{out}^{sup}$	78.7%
$\mathcal{L}_{in}^{sup}$	67.4%

Table 1: ImageNet Top-1 classification accuracy for supervised contrastive losses on ResNet-50 for a batch size of 6144.

Top-1 classification accuracy

Loss	Architecture	Augmentation	Top-1	Top-5
Cross-Entropy (baseline)	ResNet-50	MixUp [61]	77.4	93.6
Cross-Entropy (baseline)	ResNet-50	CutMix [60]	78.6	94.1
Cross-Entropy (baseline)	ResNet-50	AutoAugment [5]	78.2	92.9
Cross-Entropy (our impl.)	ResNet-50	AutoAugment [30]	77.6	95.3
SupCon	ResNet-50	AutoAugment [5]	**78.7**	**94.3**
Cross-Entropy (baseline)	ResNet-200	AutoAugment [5]	80.6	95.3
Cross-Entropy (our impl.)	ResNet-200	Stacked RandAugment [49]	80.9	95.2
SupCon	ResNet-200	Stacked RandAugment [49]	**81.4**	**95.9**
SupCon	ResNet-101	Stacked RandAugment [49]	80.2	94.7

Dataset	SimCLR[3]	Cross-Entropy	Max-Margin [32]	SupCon
CIFAR10	93.6	95.0	92.4	**96.0**
CIFAR100	70.7	75.3	70.5	**76.5**
ImageNet	70.2	78.2	78.0	**78.7**

reference

- A Simple Framework for Contrastive Learning of Visual Representations
  - https://arxiv.org/pdf/2002.05709.pdf
- Supervised Contrastive Learning
  - https://arxiv.org/pdf/2004.11362.pdf