Supervised Contrastive Learning

20. 04. 11, arXiv

Presented by Junsoo Lee

2020. 05. 07

KAIST

DAVIAN

Data and Visual Analytics Lab

Learning Representation through Contrastive Learning

Supervised

- Baseline: Cross-Entropy
- Label smoothing
- Mixup: data augmentation
- Self-distillation

Unsupervised

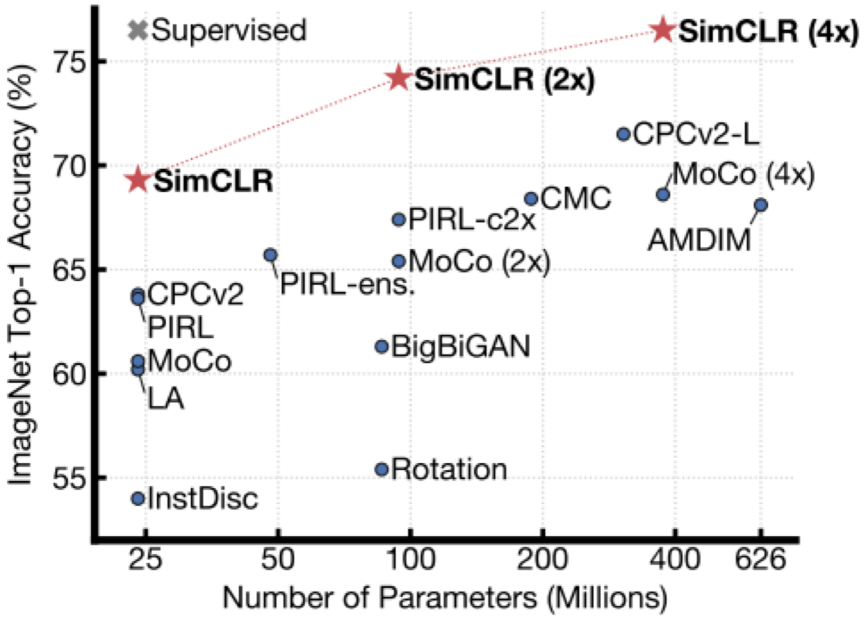
- Baseline: Auto Encoder
- Contrastive Learning

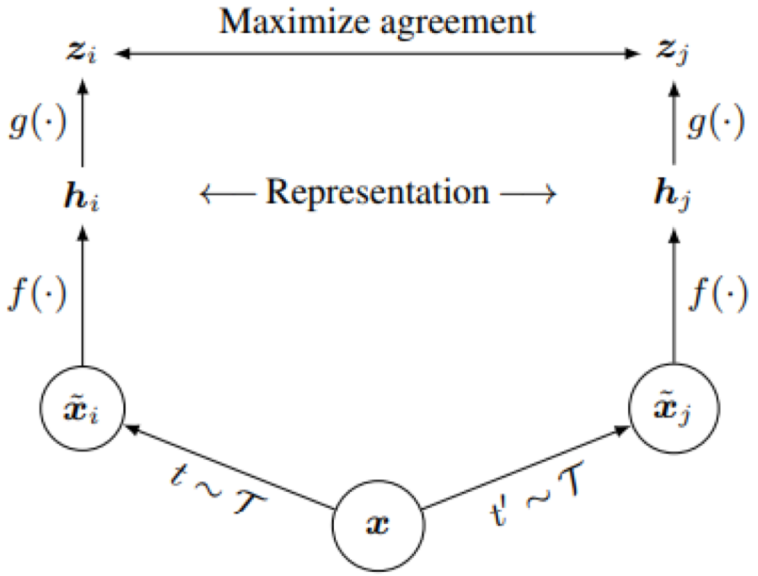
- These modifications commonly aim to learn improved “generalization”, “robustness” and “calibration”.

Learning Representation through Contrastive Learning

A Simple Framework for Contrastive Learning of Visual Representations

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





Extension of Contrastive Learning

Supervised Contrastive Learning

Prannay Khosla*
Google Research
prannayk@google.com

Piotr Teterwak*
Google Research
pteterwak@google.com

Chen Wang
Google Research
wangch@google.com

Aaron Sarna
Google Research
sarna@google.com

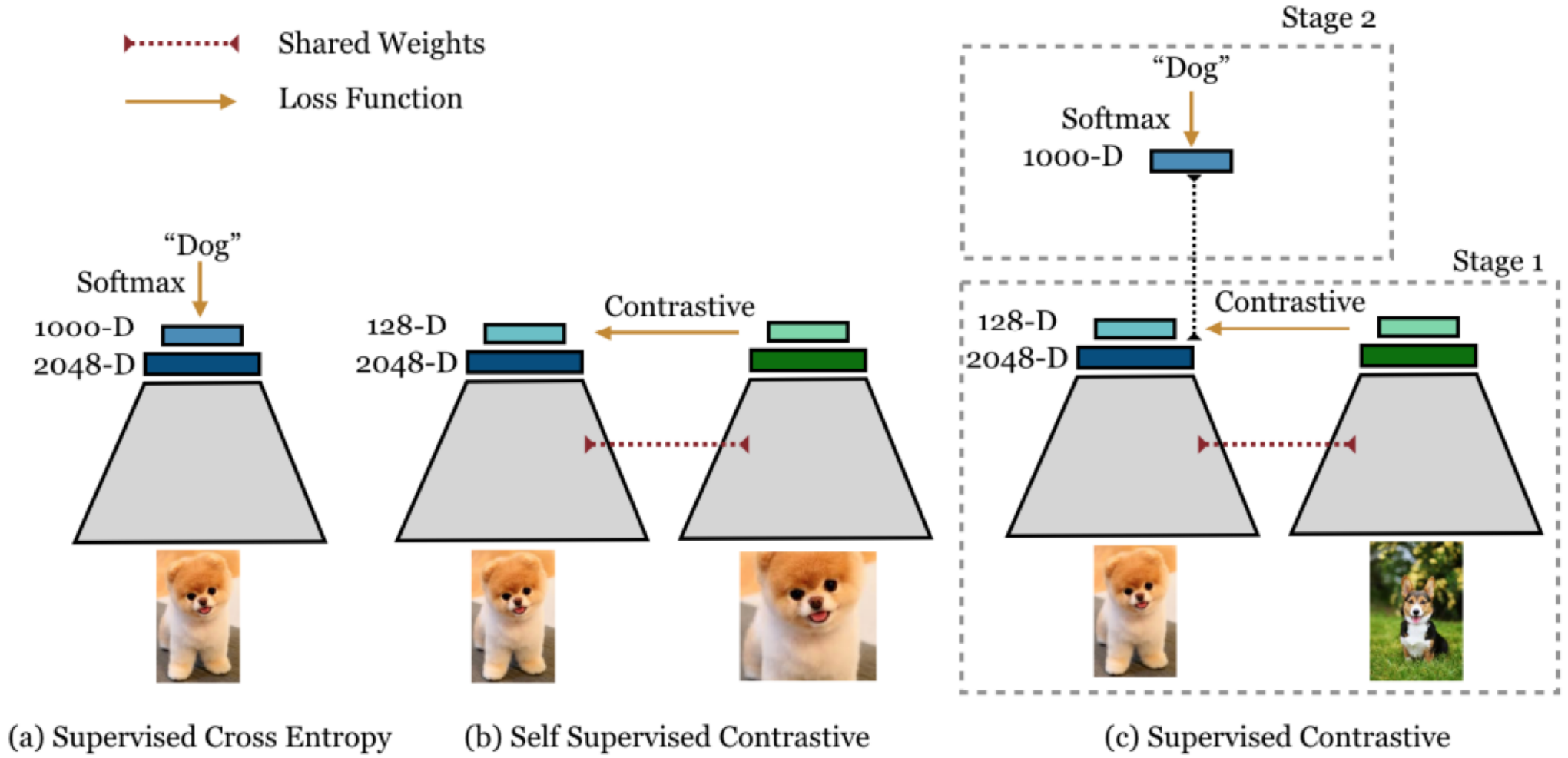
Yonglong Tian
MIT
yonglong@mit.edu

Phillip Isola
MIT
phillipi@mit.edu

Aaron Maschinot
Google Research
amaschinot@google.com

Ce Liu
Google Research
celiu@google.com

Overview:



(a) Supervised Cross Entropy (b) Self Supervised Contrastive (c) Supervised Contrastive

Distinctions from previous approaches

- Key technical novelty in this work is to consider **“many positives”** per anchor in addition to many negatives (as opposed to the convention in self-supervised contrastive learning which uses only a single positive.
- We use provided labels to select the positives and negatives.
- By using many positives and many negatives, we are able to better model both *intra-class* and *inter-class* variability.

Distinctions from previous approaches

- Triplet loss, Contrastive learning, Supervised contrastive learning (proposed)
- They consist of two **opposing forces**: for a given *anchor* point, the first force pulls the anchor closer in representation space to other points, and the second force pushes the anchor farther away from other points.
- The former set is known as *positives*, and the latter as *negatives*.

$$Loss = \sum_{i=1}^{N} \left[\|f_i^a - f_i^p\|_2^2 - \|f_i^a - f_i^n\|_2^2 + \alpha \right]_+$$

$$\mathcal{L}^{sup} = \sum_{i=1}^{2N} \mathcal{L}_i^{sup}$$

$$\mathcal{L}^{self} = \sum_{i=1}^{2N} \mathcal{L}_i^{self}$$

$$\mathcal{L}_i^{sup} = \frac{-1}{2N_{\tilde{\boldsymbol{y}}_i} - 1} \sum_{j=1}^{2N} \mathbb{1}_{i \neq j} \cdot \mathbb{1}_{\tilde{\boldsymbol{y}}_i = \tilde{\boldsymbol{y}}_j} \cdot \log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_j / \tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{i \neq k} \cdot \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_k / \tau\right)}$$

$$\mathcal{L}_i^{self} = -\log \frac{\exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_{j(i)} / \tau\right)}{\sum_{k=1}^{2N} \mathbb{1}_{i \neq k} \cdot \exp\left(\boldsymbol{z}_i \cdot \boldsymbol{z}_k / \tau\right)}$$

Experiment #1 : ImageNet Classification Accuracy

Loss	Architecture	Top-1	Top-5
Cross Entropy (baselines)	AlexNet [27]	56.5	84.6
	VGG-19+BN [42]	74.5	92.0
	ResNet-18 [20]	72.1	90.6
	MixUp ResNet-50 [56]	77.4	93.6
	CutMix ResNet-50 [55]	78.6	94.1
	Fast AA ResNet-50 [9]	77.6	95.3
	Fast AA ResNet-200 [9]	80.6	95.3
Cross Entropy (our implementation)	ResNet-50	77.0	92.9
	ResNet-200	78.0	93.3
Supervised Contrastive	ResNet-50	**78.8**	**93.9**
	ResNet-200	**80.8**	**95.6**

Table 1: Top-1/Top-5 accuracy results on ImageNet on ResNet-50 and ResNet-200 with AutoAugment [9] being used as the augmentation for Supervised Contrastive learning. Achieving 78.8% on ResNet-50, we outperform all of the top methods whose performance is shown above. Baseline numbers are taken from the referenced papers and we also additionally re-implement cross-entropy ourselves for fair comparison.

- Measured by linear evaluation protocol.

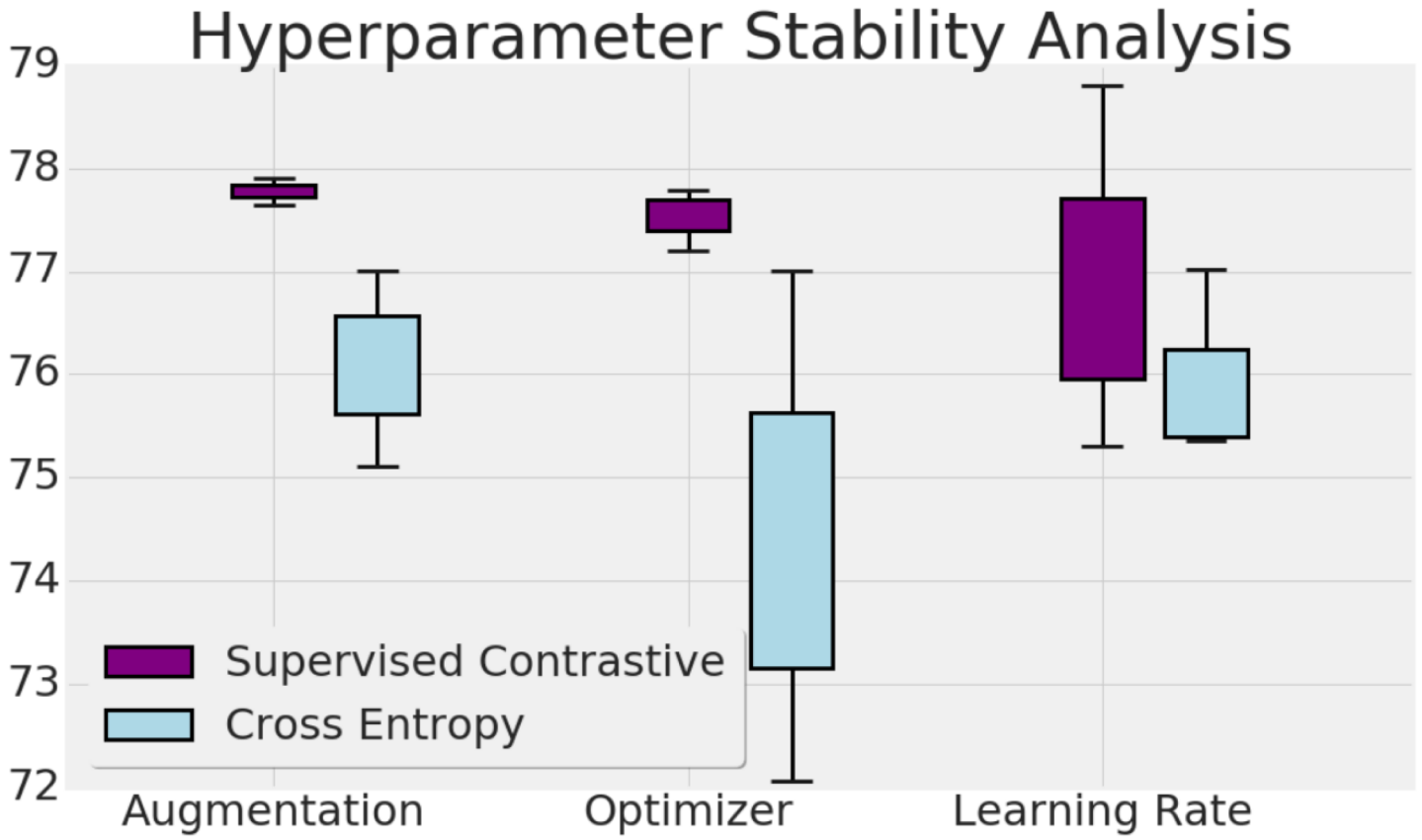
Experiment #2 : Robustness to Image Corruptions

Loss	Architecture	rel. mCE	mCE
Cross Entropy (baselines)	AlexNet [27]	100.0	100.0
	VGG-19+BN [42]	122.9	81.6
	ResNet-18 [20]	103.9	84.7
Cross Entropy (our implementation)	ResNet-50	103.7	68.4
	ResNet-200	96.6	69.4
Supervised Contrastive	ResNet-50	**87.5**	**64.4**
	ResNet-200	**77.1**	**57.2**

Table 2: Training with Supervised Contrastive Loss makes models more robust to corruptions in images, as measured by Mean Corruption Error (mCE) and relative mCE over the ImageNet-C dataset [22] (lower is better).

- DNN often lack robustness to out of distribution data or natural corruptions.
- This has been shown not only with adversarially constructed examples, but also with naturally occurring variations such as noise, blur, and JPEG compression.

Experiment #3 : Hyperparameter Stability



- Augmentations: (RandAugment, AutoAugment, SimAugment)
- Optimizers: (LARS, SGD with Momentum and RMSProp)

Experiment #4 : Effect of the number of Positives

Number of positives	1 [6]	2	3	5
Top-1 Accuracy	69.3	78.1	78.2	78.8

Table 3: Comparison of Top-1 accuracy variability as a function of the number of positives $N_{\tilde{\boldsymbol{y}}_i}$ in Eq. 4 varies from 1 to 5. Adding more positives benefits the final Top-1 accuracy. We compare against previous state of the art self-supervised work [6] which has used **one** positive which is another data augmentation of the *same sample*; see text for details.

Training Details:

- The supervised contrastive loss was trained for up to 700 epochs during the pretraining stage.
- Each training step is about 50% slower than cross-entropy.
- Aurhors trained our models with batch sizes of up to 8192, although batch sizes of 2048 suffice for most purposes for both supervised contrastive and cross entropy losses.
- All our results used a temperature of $\tau = 0.07$ and note that smaller temperature benefit training more than higher ones. But, lower temperatures can be sometimes harder to train due to numerical stability issues.
- Authors get the best performance for supervised contrastive loss by using LARS for pre-training and RMSProp for training the dense layer on the top of the frozen network.

Appendix:

```
...reaserverone: ~ (ssh) ⌘1 | ...rverone: ~ (bash) ⌘2 | ...aserverone: ~ (bash) ⌘3
    self.writer.add_scalar('validation_accuracy', val_acc, global_step=valid_n_iter)
NameError: name 'val_acc' is not defined
(base) junsulee@koreaserverone:/home/userB/junsulee/SimCLR$ CUDA_VISIBLE_DEVICES=0 python run.py
Please install apex for mixed precision training from: https://github.com/NVIDIA/apex
Please install apex for mixed precision training from: https://github.com/NVIDIA/apex
Running on: cuda
Feature extractor: resnet18
Files already downloaded and verified
num_train: 5000, train_idx4750, valid_idx: 250
74 3
 Ready for data loaders..
 Start to train..
[Epoch: 0/80][Iters: 0/74=>0] >> Loss : 2.3780617713928223, Acc: 7.8125
[Epoch: 0/80][Iters: 10/74=>10] >> Loss : 2.015319585800171, Acc: 28.125
[Epoch: 0/80][Iters: 20/74=>20] >> Loss : 1.793121576309204, Acc: 28.125
[Epoch: 0/80][Iters: 30/74=>30] >> Loss : 1.8449662923812866, Acc: 28.125
[Epoch: 0/80][Iters: 40/74=>40] >> Loss : 1.9399073123931885, Acc: 21.875
[Epoch: 0/80][Iters: 50/74=>50] >> Loss : 1.9184147119522095, Acc: 20.3125
[Epoch: 0/80][Iters: 60/74=>60] >> Loss : 1.8186004161834717, Acc: 25.0
[Epoch: 0/80][Iters: 70/74=>70] >> Loss : 1.7424849271774292, Acc: 29.6875
 Start to eval..
```

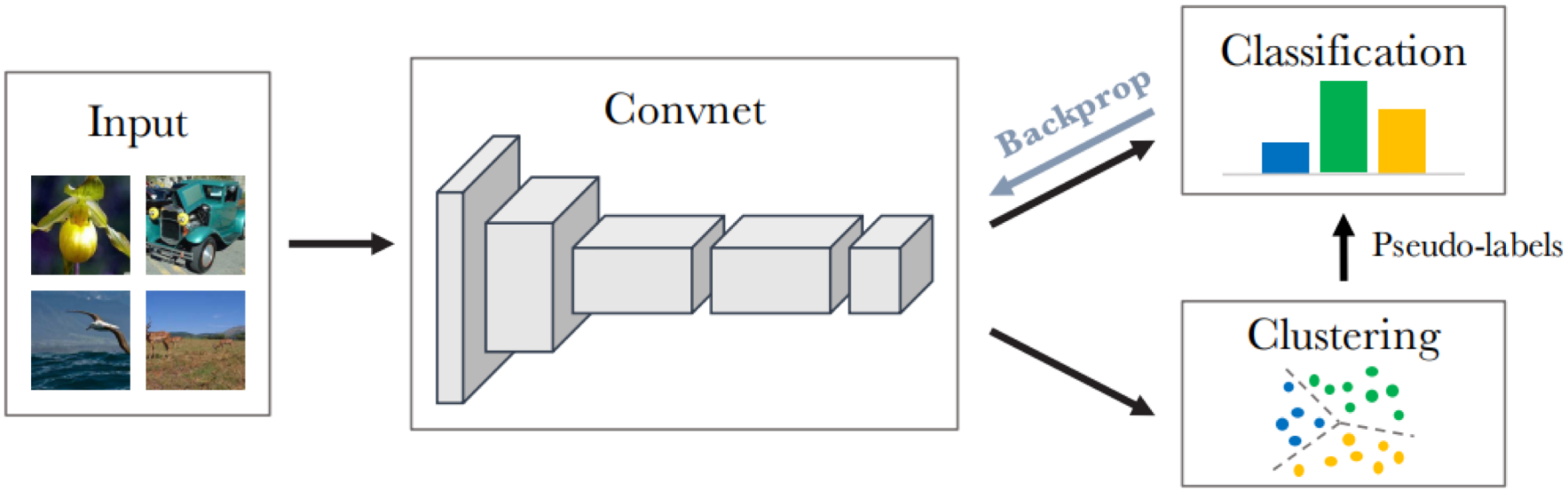
```
(base) junsulee@koreaserverone:/home/userB/junsulee/SimCLR$ python linear_evalua
Running on: cuda
{'ckpt_path': PosixPath('/home/userB/junsulee/SimCLR/logs/05-05/resnet18/b512_de
 1, 'valid_size': 0.05}, 'desc': 'b512_debug', 'epochs': 80, 'eval_every_n_epoch
/userB/junsulee/SimCLR/logs/05-05/resnet18/b512_debug', 'loss': {'temperature':
'/home/userB/junsulee/SimCLR/logs/05-05/resnet18/b512_debug/tensorboard', 'weigh
Feature extractor: resnet18
 Load SimCLR model done..
 Init Linear model done..
Files already downloaded and verified
num_train: 5000, train_idx4750, valid_idx: 250
74 3
 Start to train..
[Epoch: 0/80][Iters: 0/74=>0] >> Loss : 2.532953977584839, Acc: 7.8125
[Epoch: 0/80][Iters: 10/74=>10] >> Loss : 2.2594380378723145, Acc: 14.0625
[Epoch: 0/80][Iters: 20/74=>20] >> Loss : 1.834546685218811, Acc: 37.5
[Epoch: 0/80][Iters: 30/74=>30] >> Loss : 1.6400777101516724, Acc: 51.5625
[Epoch: 0/80][Iters: 40/74=>40] >> Loss : 1.5294312238693237, Acc: 60.9375
[Epoch: 0/80][Iters: 50/74=>50] >> Loss : 1.3461750745773315, Acc: 67.1875
[Epoch: 0/80][Iters: 60/74=>60] >> Loss : 1.2067757844924927, Acc: 65.625
[Epoch: 0/80][Iters: 70/74=>70] >> Loss : 1.1077724695205688, Acc: 67.1875
 Start to eval..
```

Learning Representation through Contrastive Learning

Deep Clustering for Unsupervised Learning of Visual Features

Mathilde Caron, Piotr Bojanowski, Armand Joulin, and Matthijs Douze

Facebook AI Research
{mathilde,bojanowski,ajoulin,matthijs}@fb.com



Learning Representation through Contrastive Learning

- Motivation:
  - Contrastive Learning 을 통해 label 없이도 "어느정도" 유의미한 feature 를 만들어낸다는 관찰결과와
  - 불완전한 feature 로 부터 Clustering 을 통해서 얻어낸 Pseudo-labels 를 할당한 뒤 학습에 활용 하는 방법이 효과가 있음을 보인 이전 관찰결과가 존재하니
  - 이를 결합하여 시너지 효과가 있는지 알아보면 어떨까
- Expected Contribution:
  - Unsupervised method 중 SOTA 를 달성하자
  - 일부 Noise label setting 에서 supervised setting 보다 우수함을 보이자
  - Limited-label setting 에서 supervised 보다 우수함을 보이자.