Rethinking Data Augmentation: Self-Supervision and Self-Distillation

Hankook Lee, Sung Ju Hwang, Jinwoo Shin

Korea Advanced Institute of Science and Technology

ICML 2020

Cite : 8

2020.07.08

임진혁

0 Index

1 Introduction: 기존방법론문제제기

Abstract

- 기존 Data Augmentation은 (이후 DA 약칭) 학습데이터를 늘림으로써 보편적 성능을 향상시키는데 널리 쓰이고 있다. 이러한 augmentation data set의 모든 라벨은 source 데이터의 라벨을 따른다.

  해당 연구는 이 부분에 문제를 제기한다.

- 앞서 말한 "라벨의 종속" 즉 augmentation data의 <<label invariance>>을 강제한 학습은 오히려 성능을 감소시킬 수 있다. 해당 연구는 이를 해결하기 위한 "Joint Learning"과 "Aggregated Inference"을 제안한다.

기존의 Data augmentation techniques에는flipping, cropping 등이 흔히 사용된다.
이러한 augmentation은 데이터 도메인에 따라 자칫 augmentation 결과물이 너무 넓은 분포값을 가지고 Original label의 특성을 잃음으로써 학습과 성능에 방해가 될 수 있다.

해당 연구는 이러한 한계점을 극복하기 위한 것으로

DATA AUGMENTATION , MULTI-TASK(=Self-supervision) 와의 비교 실험을 통해 그 효과를 확인했으며

1) 데이터 부족 문제 , 2)Few –shot Classification ,3) Imbalanced classification 등 분류데이터셋에서 큰 강점을 보였다.

Data Augmentation

- Sample 개수가 적을 때 NN이 쉽게 overfit 되는 한계점을 극복하기 위한 가장 대표적인 regularization 기법
- 간단하게 예를 들자면, “새의 종류”를 예측하는 task에서 색깔에 대한 augmentation은 치명적일 수 있다.

➔ **Force in-variance**

MULTI-TASK(=Self-supervision)

Self-Supervised Learning: **pretext task로 NN을 pretrain하여 downstream task로 transfer learning.**

pretext task : Unlabeled 데이터들을 이용하여 사용자가 새로운 문제를 정의하여 이에 대한 정답을
Self- supervised label이라 하며 이 때의 새로운 문제를 뜻함.

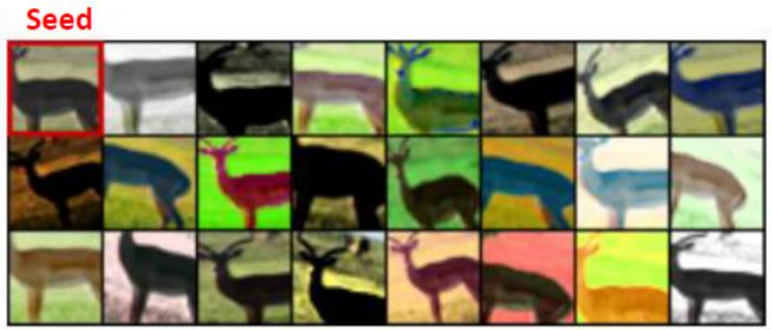
DOWNSTREAM TASK: Pretrain된 가중치를 사용하여 원하는 테스크에 fine-tune

self- supervised 학습은 일종의 비지도학습으로 라벨이 없는 데이터를 해당 데이터의 구조나 특성을 기반으로 라벨링하여 학습함으로써 High-level representations 학습을 가능케 한다.

1 Introduction: 기존방법론문제제기

MULTI-TASK(=Self-supervision)

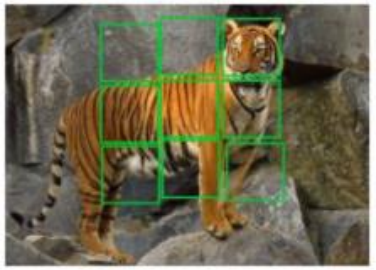
Exemplar, 2014 NIPS



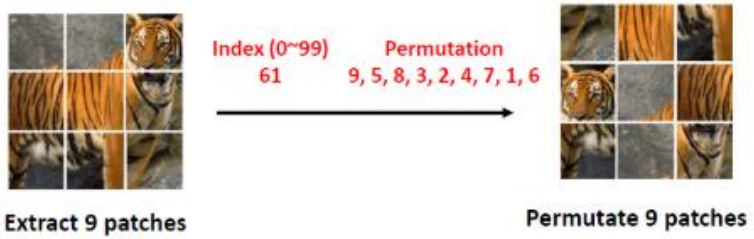
Train with STL-10 dataset (96x96)

[Exemplar]

Jigsaw Puzzle, 2016 ECCV



Sample image

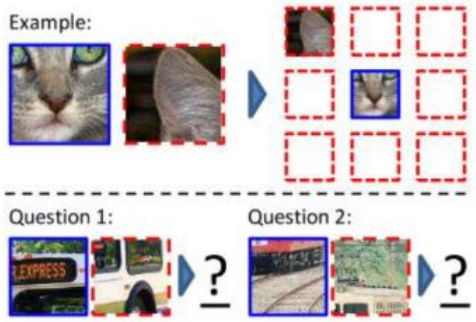


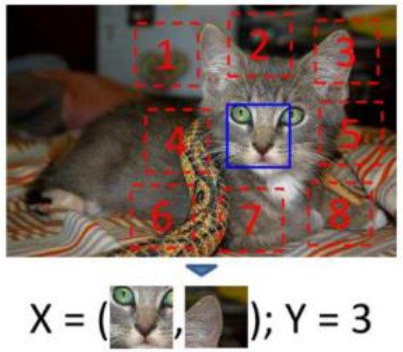
Extract 9 patches

Permutate 9 patches

[Jigsaw Puzzle]

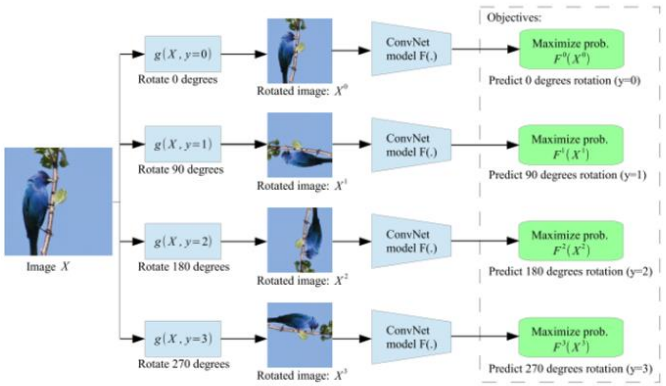
Context Prediction. 2015 ICCV





[Context Prediction]

Rotation, 2018 ICLR



[Rotation]

➔ **Force in-variance**

MULTI-TASK(=Self-supervision)

Multi-task, 2017 ICCV

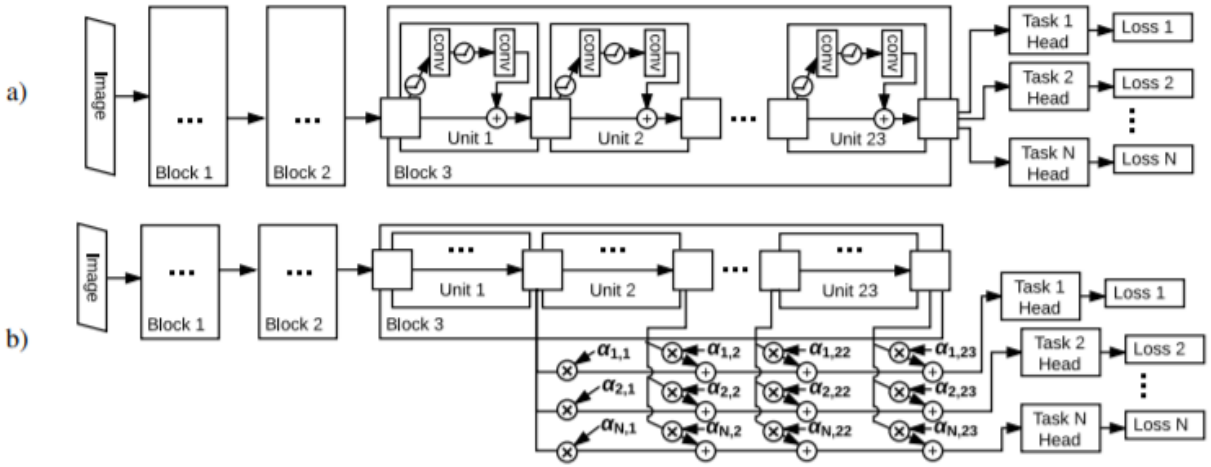


Figure 1. The structure of our multi-task network. It is based on ResNet-101, with block 3 having 23 residual units. a) Naive shared-trunk approach, where each "head" is attached to the output of block 3. b) the lasso architecture, where each "head" receives a linear combination of unit outputs within block3, weighted by the matrix α, which is trained to be sparse.

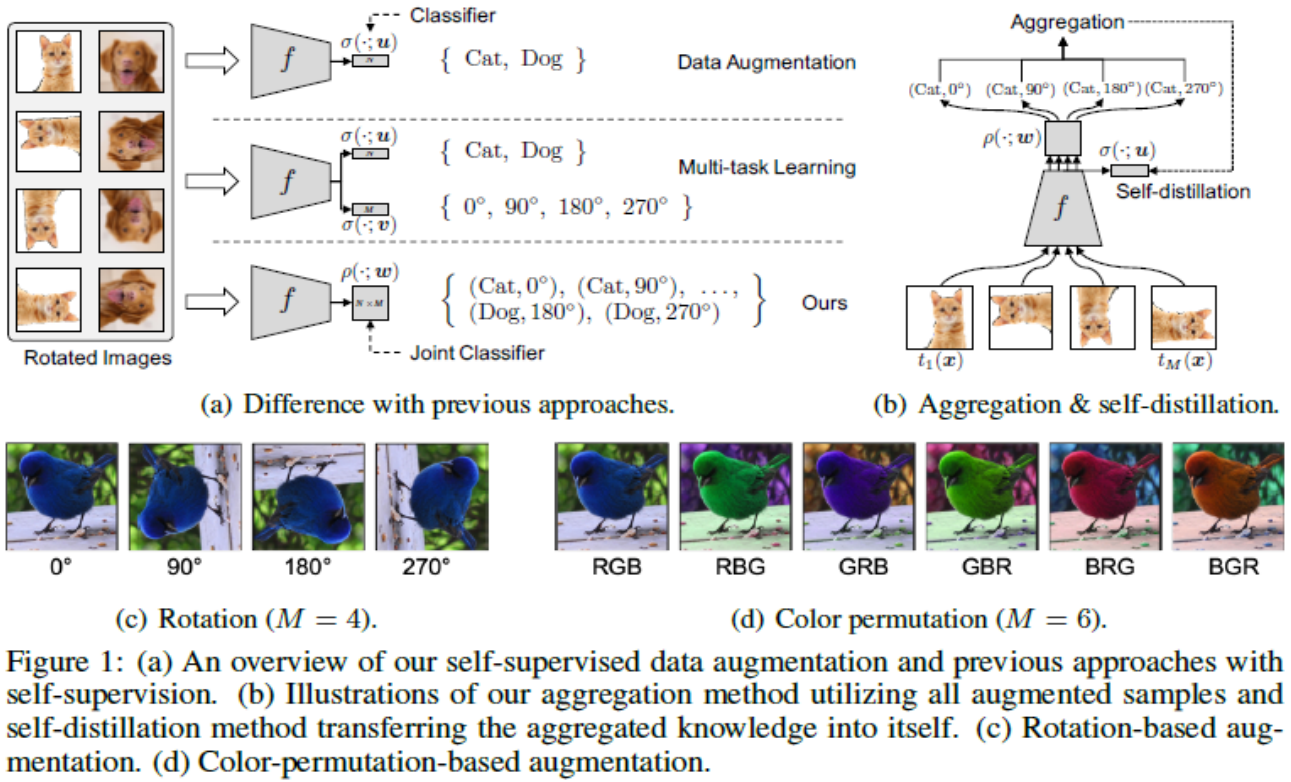
각각의 original label를 예측하는 classifier가 여전히 존재 + self -label의 문제점

➔ **Force in-variance**

2 Related work

제안하는 방법론 :

"Our main idea is simple and intuitive :
maintain a single joint classifier, instead of two separate classifiers"



(a) Difference with previous approaches.

(b) Aggregation & self-distillation.

(c) Rotation ($M = 4$).

(d) Color permutation ($M = 6$).

Figure 1: (a) An overview of our self-supervised data augmentation and previous approaches with self-supervision. (b) Illustrations of our aggregation method utilizing all augmented samples and self-distillation method transferring the aggregated knowledge into itself. (c) Rotation-based augmentation. (d) Color-permutation-based augmentation.

(기존의 방법론이면)
original and self-supervised tasks are CIFAR10 (10 labels) and rotation (4 labels)

(joint probability distribution 학습)
all possible combinations of 40 labels

-> no relationship between the original and self-supervised labels

➔ **X Force in-variance**

2 Related work

Notations

X: INPUT

y: $y \in \{1, \ldots, N\}$

Classifier(soft-max): $\sigma(\cdot; u)$

Augmentation input: $z = f(x; \theta)$

Data augmentation.

$$\mathcal{L}_{\text{DA}}(x, y; \theta, u) = \mathbb{E}_{t \sim T}\left[\mathcal{L}_{\text{CE}}(\sigma(f(\tilde{x}; \theta); u), y)\right]$$

➔ **Force in-variance**

➔ forces the classifier invariant to transformations.

Self-supervision

기본 classification 로스(CE)와 self-supervised task의 로스(CE)의 합 M: self-supervised labels

$$\mathcal{L}_{\text{MT}}(x, y; \theta, u, v) = \frac{1}{M}\sum_{j=1}^{M} \mathcal{L}_{\text{CE}}(\sigma(f(\tilde{x}_j; \theta); u), y) + \mathcal{L}_{\text{CE}}(\sigma(f(\tilde{x}_j; \theta); v), j)$$

➔ **Force in-variance**

➔ forces the classifier invariant to transformations.

3 Proposed Approach

(1) ELIMINATING INVARINCE VIA JOINT-LABEL CLASSIFIER

결국 해당 방법론의 핵심은 불필요한 < Invariant property>를 제거하는 것!

$$\mathcal{L}_{\mathrm{SDA}}(\boldsymbol{x}, y; \boldsymbol{\theta}, \boldsymbol{w}) = \frac{1}{M} \sum_{j=1}^{M} \mathcal{L}_{\mathrm{CE}}(\rho(f(\tilde{\boldsymbol{x}}_j; \boldsymbol{\theta}); \boldsymbol{w}), (y, j))$$

새로운 Notation: joint softmax classifier

$\rho(\cdot; \boldsymbol{w})$: Joint probability를 반영할 수 있음!

$$\tilde{P}(i, j|\tilde{\boldsymbol{x}}) = \rho_{ij}(\tilde{\boldsymbol{z}}; \boldsymbol{w}) = \exp(\boldsymbol{w}_{ij}^{\top}\tilde{\boldsymbol{z}}) / \sum_{k,l} \exp(\boldsymbol{w}_{kl}^{\top}\tilde{\boldsymbol{z}}).$$

3 Proposed Approach

(2) Aggregated inference.

여기서 드는 의문: 그러면 10(label) , 10(transformation)이면 라벨이 10개에서 100개로 늘어나는 것 아닌가?
단순히 Joint Label을 만들다고 성능이 증가할 수 있는가?

→ 우리는 Transformation을 이미 알고있으므로 고정시킬 수 있다.

$$P(i|\tilde{\boldsymbol{x}}_j, j) = \exp(\boldsymbol{w}_{ij}^\top \tilde{\boldsymbol{z}}_j) / \sum_k \exp(\boldsymbol{w}_{kj}^\top \tilde{\boldsymbol{z}}_j) \text{ where } \tilde{\boldsymbol{z}}_j = f(\tilde{\boldsymbol{x}}_j; \boldsymbol{\theta}).$$

→ 이를 통해, single 모델이지만 동일한 라벨(original)에 대해 여러 모델(transformation)의 답을 평균화한 것과 동일해져서 앙상블과 동일한 효과를 볼 수 있다

$$P_{\text{aggregated}}(i|x) = \frac{\exp(s_i)}{\sum_{k=1}^{N} \exp(s_k)} \quad \text{where} \quad s_i = \frac{1}{M} \sum_{j=1}^{M} \boldsymbol{w}_{ij}^\top \tilde{\boldsymbol{z}}_j.$$

(2) Aggregated inference.

$$P(i|\tilde{\boldsymbol{x}}_j, j) = \exp(\boldsymbol{w}_{ij}^\top \tilde{\boldsymbol{z}}_j) / \sum_k \exp(\boldsymbol{w}_{kj}^\top \tilde{\boldsymbol{z}}_j) \text{ where } \tilde{\boldsymbol{z}}_j = f(\tilde{\boldsymbol{x}}_j; \boldsymbol{\theta}).$$

$$P(i|\tilde{\boldsymbol{x}}_j, j) = \exp(\boldsymbol{w}_{ij}^\top \tilde{\boldsymbol{z}}_j) / \sum_k \exp(\boldsymbol{w}_{kj}^\top \tilde{\boldsymbol{z}}_j) \text{ where } \tilde{\boldsymbol{z}}_j = f(\tilde{\boldsymbol{x}}_j; \boldsymbol{\theta}).$$

3 Proposed Approach

(3) Self-distillation from aggregation.

하지만 해당 방법은 여전히 문제점이 있다.
바로 aggregated inference를 위해서 총 M 배의 COST가 추가로 소요된다는 것

이 부분에 대한 COST를 줄이고 Inference 속도를 높이기 위해 self- distill을 적용하였다.

$$\mathcal{L}_{\text{SDA+SD}}(x, y; \theta, w, u) = \mathcal{L}_{\text{SDA}}(x, y; \theta, w) + D_{\text{KL}}(P_{\text{aggregated}}(\cdot|x) \| \sigma(f(x; \theta); u)) + \beta \mathcal{L}_{\text{CE}}(\sigma(f(x; \theta); u), y)$$

T는 ??

S는 ??

3 Proposed Approach

(3) Self-distillation from aggregation.

하지만 해당 방법은 여전히 문제점이 있다.
바로 aggregated inference를 위해서 총 M 배의 COST가 추가로 소요된다는 것

이 부분에 대한 COST를 줄이고 Inference 속도를 높이기 위해 self- distill을 적용하였다.

$$\mathcal{L}_{\text{SDA+SD}}(x, y; \theta, w, u) = \mathcal{L}_{\text{SDA}}(x, y; \theta, w) + D_{\text{KL}}(P_{\text{aggregated}}(\cdot|x) \| \sigma(f(x; \theta); u)) + \beta \mathcal{L}_{\text{CE}}(\sigma(f(x; \theta); u), y)$$

T: $P_{\text{aggregated}}(\cdot|x)$

S: $\sigma(f(x; \theta); u)$

1차 학습은 어떻게 할까?

이런 Distill이 어떻게 추론을 빠르게 하는 걸까?

Experiments

1) 전반적인 성능 : CIFAR10/100/ tiny-ImageNet

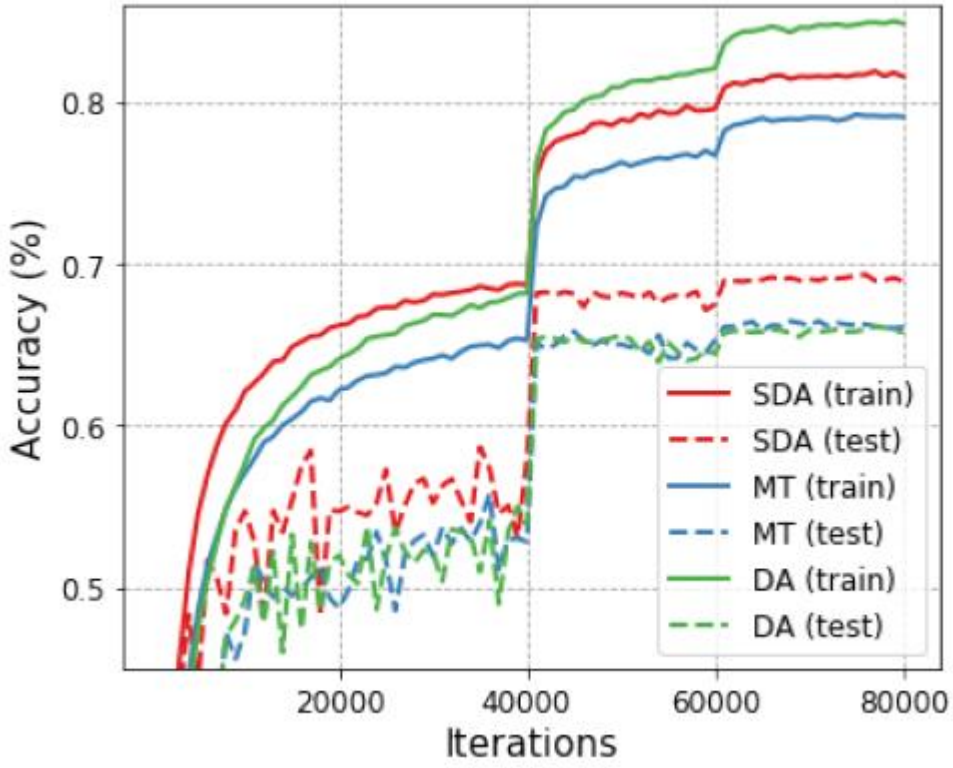


Table 1: Classification accuracy (%) of single inference using data augmentation (DA), multi-task learning (MT), and ours self-supervised data augmentation (SDA) with rotation. The best accuracy is indicated as bold, and the relative gain over the baseline is shown in brackets.

Dataset	Baseline	DA	MT	SDA+SI
CIFAR10	92.39	90.44 (-2.11%)	90.79 (-1.73%)	**92.50** (+0.12%)
CIFAR100	68.27	65.73 (-3.72%)	66.10 (-3.18%)	**68.68** (+0.60%)
tiny-ImageNet	63.11	60.21 (-4.60%)	58.04 (-8.03%)	**63.99** (+1.39%)

Experiments

2) Aggregated inference 성능 비교 : CIFAR10/100/ tiny-ImageNet

Table 2: Classification accuracy (%) of the ten-crop, indenpendent ensemble, and our aggregation using rotation (SDA+AG). The best accuracy is indicated as bold, and the relative gain over the baseline is shown in brackets.

	Single Model			4 Models	
Dataset	Baseline	ten-crop	SDA+AG	Ensemble	Enemble + SDA+AG
CIFAR10	92.39	93.33	**94.50** (+2.28%)	94.36	**95.10** (+2.93%)
CIFAR100	68.27	70.54	**74.14** (+8.60%)	74.82	**76.40** (+11.9%)
tiny-ImageNet	63.11	64.95	**66.95** (+6.08%)	68.18	**69.01** (+9.35%)

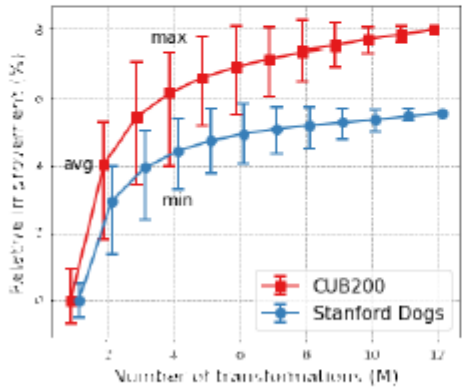


Figure 3: Relative improvements (%) of aggregation versus the number of transformations.

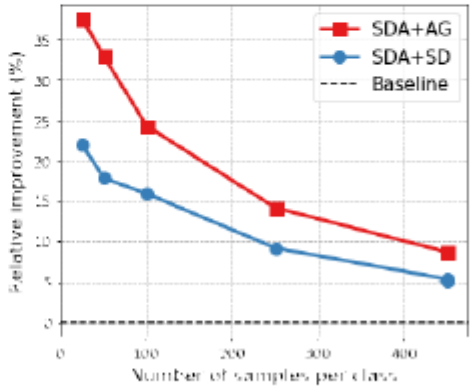


Figure 4: Relative improvements (%) over baselines on subsets of CIFAR100.

Experiments

3) Transformation에 따른 성능 비교 : 다양한 데이터셋

Table 3: Classification accuracy (%) using self-supervised data augmentation with rotation and color permutation. SDA+SD and SDA+AG indicate the single inference trained by $\mathcal{L}_{\text{SDA+SD}}$, and the aggregated inference trained by $\mathcal{L}_{\text{SDA}}$, respectively. The relative gain is shown in brackets.

		Rotation		Color Permutation	
Dataset	Baseline	SDA+SD	SDA+AG	SDA+SD	SDA+AG
CIFAR10	92.39	93.26 (+0.94%)	94.50 (+2.28%)	91.51 (-0.95%)	92.51 (+0.13%)
CIFAR100	68.27	71.85 (+5.24%)	74.14 (+8.60%)	68.33 (+0.09%)	69.14 (+1.27%)
CUB200	54.24	62.54 (+15.3%)	64.41 (+18.8%)	60.95 (+12.4%)	61.10 (+12.6%)
MIT67	54.75	63.54 (+16.1%)	64.85 (+18.4%)	60.03 (+9.64%)	59.99 (+9.57%)
Stanford Dogs	60.62	66.55 (+9.78%)	68.70 (+13.3%)	65.92 (+8.74%)	67.03 (+10.6%)
tiny-ImageNet	63.11	65.53 (+3.83%)	66.95 (+6.08%)	63.98 (+1.38%)	64.15 (+1.65%)

Experiments

4) Data Set의 한계 에 따른 성능 비교 : 1. 부족한 데이터 2. few shot 3. imbalanced

Table 5: Average classification accuracy (%) with 95% confidence intervals of 1000 5-way few-shot tasks on mini-ImageNet, CIFAR-FS, and FC100. † and ‡ indicates 4-layer convolutional and 28-layer residual networks (Zagoruyko & Komodakis, 2016), respectively. Others use 12-layer residual networks as Lee et al. (2019). The best accuracy is indicated as bold.

Method	mini-ImageNet		CIFAR-FS		FC100	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML† (Finn et al., 2017)	48.70±1.84	63.11±0.92	58.9±1.9	71.5±1.0	-	-
R2D2† (Bertinetto et al., 2019)	-	-	65.3±0.2	79.4±0.1	-	-
RelationNet† (Sung et al., 2018)	50.44±0.82	65.32±0.70	55.0±1.0	69.3±0.8	-	-
SNAIL (Mishra et al., 2018)	55.71±0.99	68.88±0.92	-	-	-	-
TADAM (Oreshkin et al., 2018)	58.50±0.30	76.70±0.30	-	-	40.1±0.4	56.1±0.4
LEO‡ (Rusu et al., 2019)	61.76±0.08	77.59±0.12	-	-	-	-
MetaOptNet-SVM (Lee et al., 2019)	62.64±0.61	78.63±0.46	72.0±0.7	84.2±0.5	41.1±0.6	55.5±0.6
ProtoNet (Snell et al., 2017)	59.25±0.64	75.60±0.48	72.2±0.7	83.5±0.5	37.5±0.6	52.5±0.6
ProtoNet + **SDA+AG** (ours)	62.22±0.69	77.78±0.51	**74.6±0.7**	**86.8±0.5**	40.0±0.6	55.7±0.6
MetaOptNet-RR (Lee et al., 2019)	61.41±0.61	77.88±0.46	72.6±0.7	84.3±0.5	40.5±0.6	55.3±0.6
MetaOptNet-RR + **SDA+AG** (ours)	**62.93±0.63**	**79.63±0.47**	73.5±0.7	86.7±0.5	**42.2±0.6**	**59.2±0.5**

Table 6: Classification accuracy (%) on imbalance datasets of CIFAR10/100. Imbalance Ratio is the ratio between the numbers of samples of most and least frequent classes. The best accuracy is indicated as bold, and the relative gain is shown in brackets.

Imbalance Ratio (N_{max}/N_{min})	Imbalanced CIFAR10		Imbalanced CIFAR100	
	100	10	100	10
Baseline	70.36	86.39	38.32	55.70
Baseline + **SDA+SD** (ours)	74.61 (+6.04%)	89.55 (+3.66%)	43.42 (+13.3%)	60.79 (+9.14%)
CB-RW (Cui et al., 2019)	72.37	86.54	33.99	57.12
CB-RW + **SDA+SD** (ours)	77.02 (+6.43%)	89.50 (+3.42%)	37.50 (+10.3%)	**61.00** (+6.79%)
LDAM-DRW (Cao et al., 2019)	77.03	88.16	42.04	58.71
LDAM-DRW + **SDA+SD** (ours)	**80.24** (+4.17%)	**89.58** (+1.61%)	**45.53** (+8.30%)	59.89 (+1.67%)

감사합니다