DMQA Open Seminar

Image Augmentation and Adversarial Learning-based Methods

2023. 07. 14

Byeongeun Ko

Data Mining & Quality Analytics Lab.

발표자 소개

❖ **고병은 (Byeongeun Ko)**

- 고려대학교 산업경영공학과 Data Mining & Quality Analytics Lab.
- M.S. Student (2022.03 ~ Present)
- 지도교수 : 김성범 교수님

❖ **Research Interest**

- Machine/Deep Learning for Smart Factory
- Reinforcement Learning

❖ **Contact**

- E-mail : byeongeun_ko@korea.ac.kr

목차

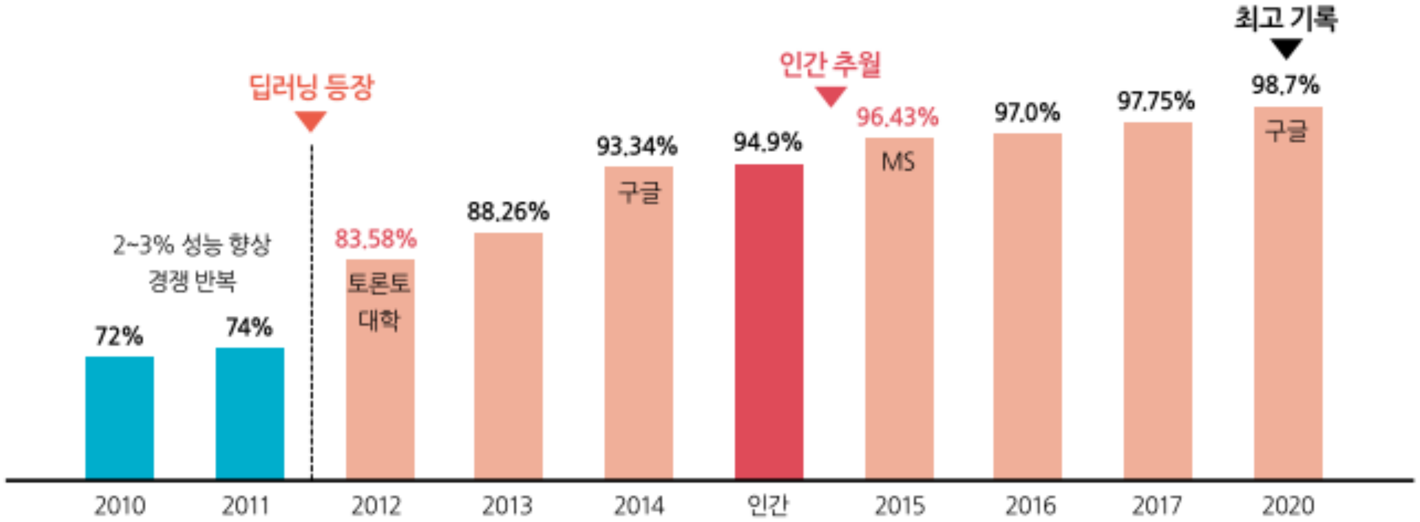
Contents

Introduction

Introduction

❖ Image Classification

- 최근 컴퓨터 비전 분야에서 딥러닝 기반 방법론은 빠르게 발전되어왔고,
  이미지 분류 문제에서 2015년에 이미 사람의 인식률을 추월



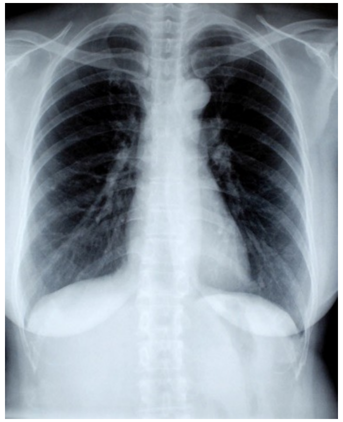
※출처: 최근 인공지능 개발 트렌드와 미래의 진화 방향, 2017, LG경제연구원
(image-net.org를 참조하여 최신 기록 정보를 추가하고 수정함)

이주열. (2020). 인공지능 이미지인식기술 동향. TTA 저널, 187, 44-51.
이승훈. (2017). 최근 인공지능 개발 트렌드와 미래의 진화 방향. LG 경제연구원, 12, 30-31.

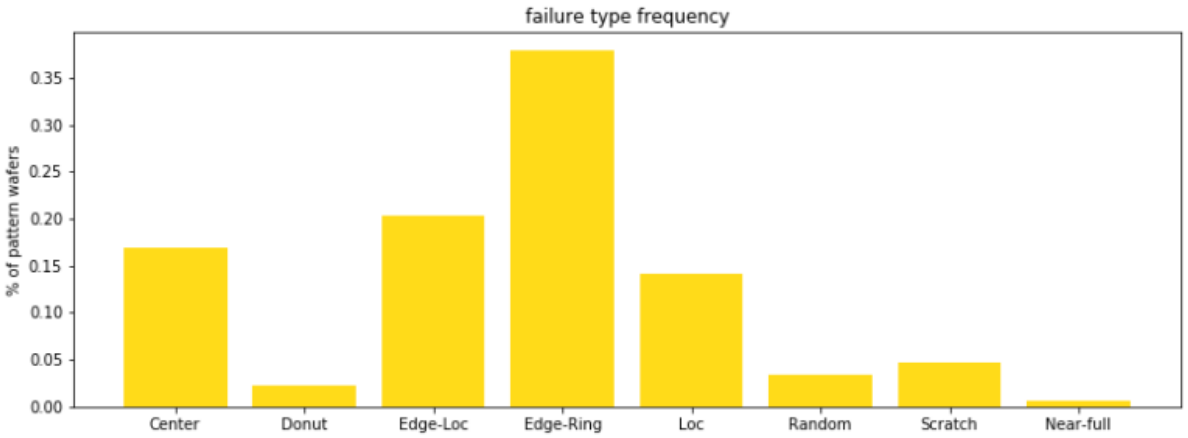
Introduction

❖ Issues of Image Classification

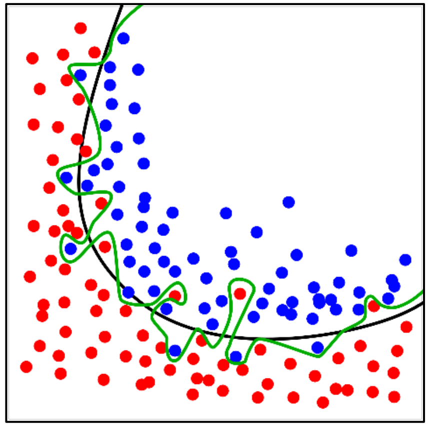
- 이러한 모델은 방대한 양의 이미지 데이터를 수집하여 학습하기에 한계점이 존재
  1. 데이터를 수집하는데 높은 비용 필요
  2. Class 별 불균형한 데이터 구성
  3. 모델이 보다 방대해지고 깊어지면서 과적합 발생, 일반화 필요



<질병 판독을 위한 X-ray 이미지>



<WM-811K Class 별 데이터 분포>



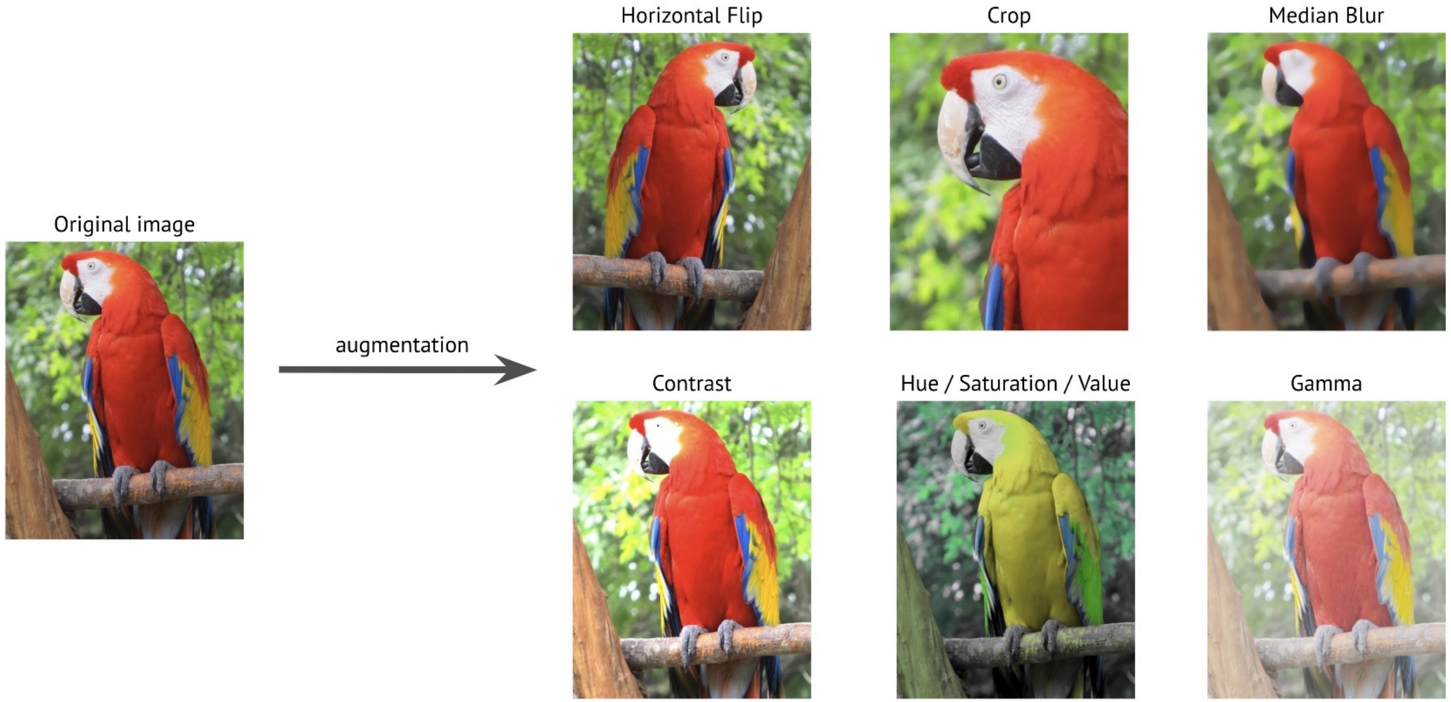
<초록선 - 과적합된 모델>

Image by kstudio on Freepik
https://ko.wikipedia.org/wiki/%EA%B3%BC%EC%A0%81%ED%95%A9

Introduction

❖ Image Augmentation

- 이미지 증강 기법이 이러한 문제를 해결하는데 활발하게 활용



<Examples of Transformations>

https://albumentations.ai/docs/introduction/image_augmentation/

Introduction

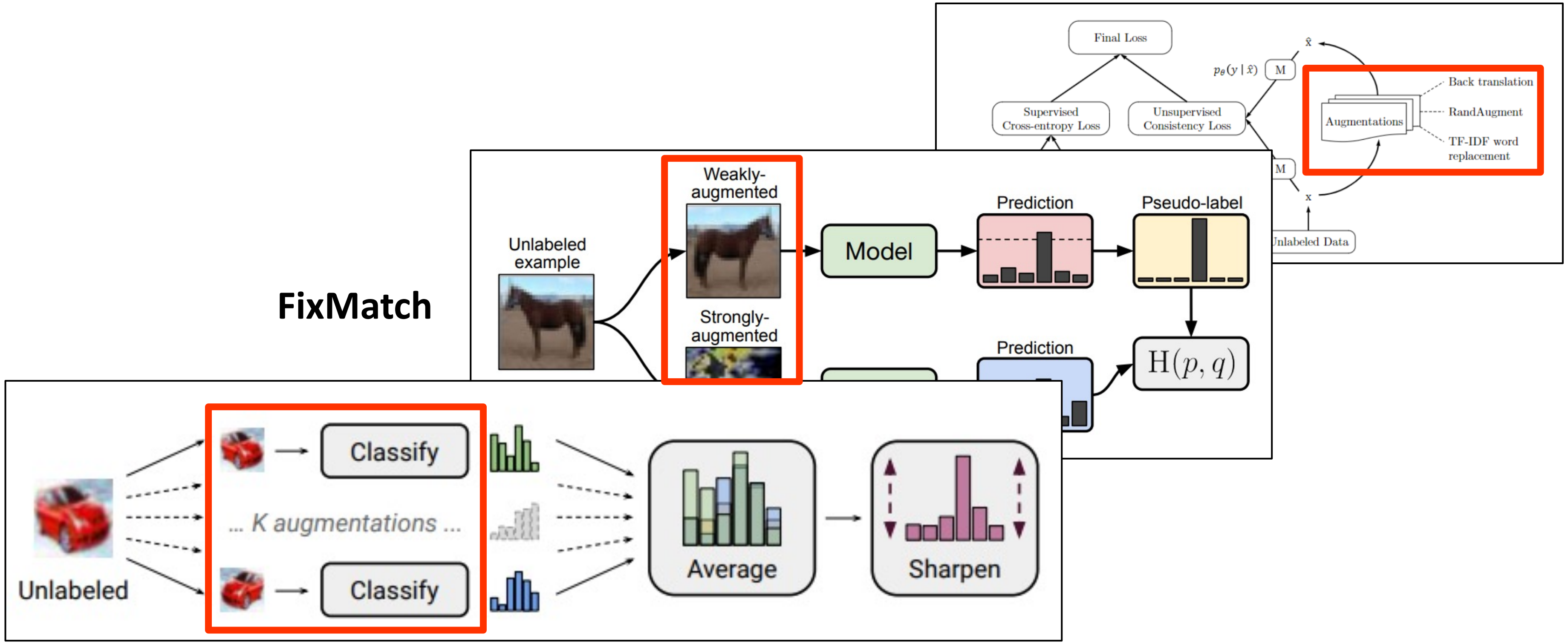
❖ Image Augmentation

- 단순히 증강하여 데이터의 수를 늘리는 것 뿐만 아니라, 최근 방법론에 다양하게 활용
- 준지도학습, 대조학습 등

UDA

FixMatch

MixMatch



Xie, Q., Dai, Z., Hovy, E., Luong, T., & Le, Q. (2020). Unsupervised data augmentation for consistency training. Advances in Neural Information Processing Systems, 33, 6256-6268.
Sohn, K., Berthelot, D., Carlini, N., Zhang, Z., Zhang, H., Raffel, C. A., ... & Li, C. L. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. Advances in neural information processing systems, 33, 596-608.
Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., & Raffel, C. A. (2019). Mixmatch: A holistic approach to semi-supervised learning. Advances in neural information processing systems, 32

Introduction

❖ How to Augment Image?

- 하지만, 이미지를 증강하는 것은 단순한 문제가 아님 → 증강 기법은 데이터에 매우 민감
  - ➢ 적절하지 않은 증강 기법은 무의미한 이미지만 생성할 수 있음
  - ➢ Domain에 따라 적용 가능한 증강 기법의 차이가 존재
- 따라서 방법론 별로 적용하는 이미지 증강 기법은 상이하며 이를 선정하고 결정하는 과정은 매우 힘듦

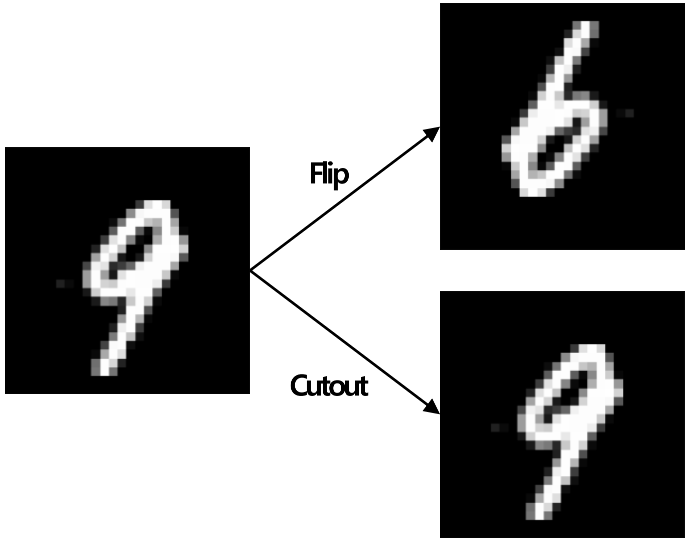


Table 1
Image augmentation algorithms used studies pertaining to image classification (up) and object detection (bottom).

Paper	Image augmentation method
AlexNet [11]	Translate, Flip, Intensity Changing
ResNet [12]	Crop, Flip
DenseNet [13]	Flip, Crop, Translate
MobileNet [14]	Crop, Elastic distortion
NasNet [15]	Cutout, Crop, Flip
ResNeSt [16]	AutoAugment, Mixup, Crop
DeiT [17]	AutoAugmentat, RandAugment, Random Erasing, Mixup, CutMix
Swin Transformer [18]	RandAugment, Mixup, CutMix, Random Erasing
Faster R-CNN [19]	Flip
YOLO [20]	Scale, Translate, Color space
SSD [21]	Crop, Resize, Flip, Color Space, Distortion
YOLOv4 [22]	Mosaic, Distortion, Scale, Color space, Crop, Flip, Rotate, Random erase, Cutout, Hide-and-Seek, GridMask, Mixup, CutMix, StyleGAN

Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A comprehensive survey of image augmentation techniques for deep learning. Pattern Recognition, 109347.

Introduction

❖ How to Augment Image?

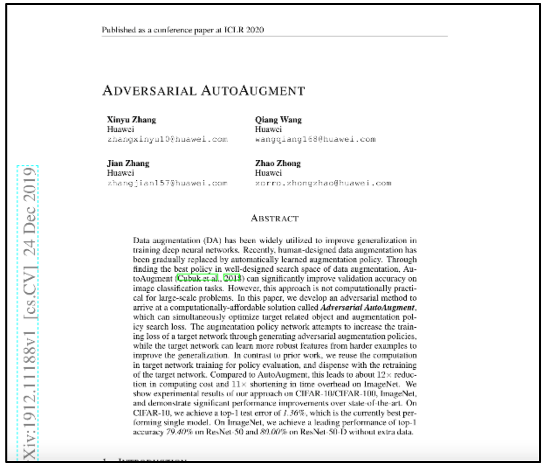
- 그렇다면, 어떻게 해야 데이터셋에 가장 최적화된 이미지 증강 기법을 적용 할 수 있을까?
- 모델이 '알아서 or 학습해서' 최적의 증강 기법이나 새로운 이미지를 생성해 줄 수 없을까?
- Survey – Taxonomy
- AA, IF-DA, SPA – Adversarial Learning-based


Pattern Recognition

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

Mingle Xu, Sook Yoon, Alvaro Fuentes, Dong Sun Park

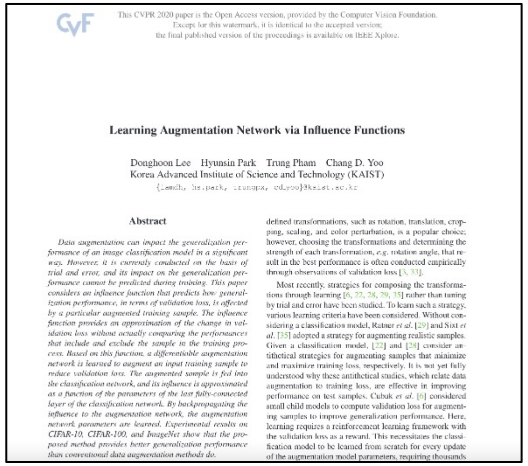
A comprehensive survey of image augmentation techniques for deep learning (2023)


Published as a conference paper at ICLR 2020

ADVERSARIAL AUTOAUGMENT

Xinyu Zhang, Huawei; Qiang Wang, Huawei; Jian Zhang, Huawei; Zhao Zhong, Huawei

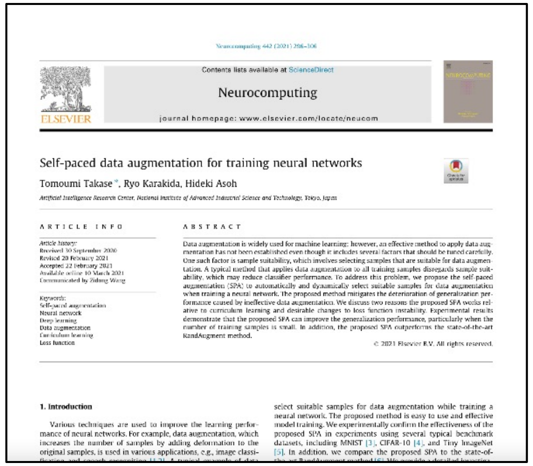
Adversarial AutoAugment (2019)


Learning Augmentation Network via Influence Functions

Donghoon Lee, Hyunsin Park, Trung Pham, Chang D. Yoo

Korea Advanced Institute of Science and Technology (KAIST)

Learning augmentation network via influence functions. (2020)


Neurocomputing

Self-paced data augmentation for training neural networks

Tomoumi Takase, Ryo Karakida, Hideki Asoh

Self-paced data augmentation for training neural networks (2021)

Paper Reviews

Paper Review

논문 리뷰

❖ A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

- 2023년, Pattern Recognition, 40회 인용

Pattern Recognition 137 (2023) 109347

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

Mingle Xu[a], Sook Yoon[b,*], Alvaro Fuentes[c], Dong Sun Park[c,*]

[a] Department of Electronics Engineering, Jeonbuk National University, Jeonbuk 54896, South Korea
[b] Department of Computer Engineering, Mokpo National University, Jeonnam 58554, South Korea
[c] Core Research Institute of Intelligent Robots, Jeonbuk National University, Jeonbuk 54896, South Korea

ARTICLE INFO

Article history:
Received 28 April 2022
Revised 22 November 2022
Accepted 15 January 2023
Available online 18 January 2023

Keywords:
Image augmentation
Deep learning
Image variation
Vicinity distribution
Data augmentation
Computer vision

ABSTRACT

Although deep learning has achieved satisfactory performance in computer vision, a large volume of images is required. However, collecting images is often expensive and challenging. Many image augmentation algorithms have been proposed to alleviate this issue. Understanding existing algorithms is, therefore, essential for finding suitable and developing novel methods for a given task. In this study, we perform a comprehensive survey of image augmentation for deep learning using a novel informative taxonomy. To examine the basic objective of image augmentation, we introduce challenges in computer vision tasks and vicinity distribution. The algorithms are then classified among three categories: model-free, model-based, and optimizing policy-based. The model-free category employs the methods from image processing, whereas the model-based approach leverages image generation models to synthesize images. In contrast, the optimizing policy-based approach aims to find an optimal combination of operations. Based on this analysis, we believe that our survey enhances the understanding necessary for choosing suitable methods and designing novel algorithms.

1. Introduction

Over the recent years, deep learning has achieved significant improvements in computer vision based on three key elements, ef-

previously [5–7], our study is characterized by several essential differences. First, we do not confine ourselves to a specific type of image, such as facial images [8]. Likewise, we consider many types of image augmentation algorithms, including generative

Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A comprehensive survey of image augmentation techniques for deep learning. Pattern Recognition, 109347.

Paper Review

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

❖ Taxonomy

- 증강 기법, 이미지 증강 알고리즘을 3개의 큰 카테고리로 분류
  - ➢ Model-free : 이미지 증강을 위해서 특별한 모델을 학습 시킬 필요 없는 경우
  - ➢ Model-based : 이미지 증강을 위해 모델 학습이 필요한 경우
  - ➢ Optimizing Policy-based : 전체 파라미터 공간(Search Space)에서 최적 증강 기법 및 파라미터를 결정

Table 2
Taxonomy with relevant methods.

Categories			Relevant methods
Model-free	Single-image	Geometrical transformation	translation, rotation, flip, scale, elastic distortion.
		Color image processing	jittering.
		Intensity transformation	blurring and adding noise, Hide-and-Seek [23], Cutout [24], Random Erasing [25], GridMask [26].
	Multiple-image	Non-instance-level	SamplePairing [27], Mixup [28], BC Learning [29], CutMix [30], Mosaic [22], AugMix [31], PuzzleMix [32], Co-Mixup [33], SuperMix [34], GridMix [35].
		Instance-level	CutPas [36], Scale and Blend [37], Context DA [38], Simple CutPas [39], Continuous CutPas [40].
Model-based	Unconditional		DCGAN [41], [42–44]
	Label-conditional		BDA [45], ImbCGAN [46], BAGAN [47], DAGAN [48], MFC-GAN [49], IDA-GAN [50].
	Image-conditional	Label-preserving	S+U Learning [51], AugGAN [52], Plant-CGAN [53], StyleAug [54], Shape bias [55].
		Label-changing	EmoGAN [56], δ-encoder [57], Debiased NN [58], StyleMix [59], GAN-MBD [60], SCIT [2].
Optimizing policy-based	Reinforcement learning-based		AutoAugment [61], Fast AA [62], PBA [63], Faster AA [64], RandAugment [65], MADAO [66], LDA [67], LSSP [68].
	Adversarial learning-based		ADA [69], CDST-DA [70], AdaTransform [71], Adversarial AA [72], IF-DA [73], SPA [74].

Paper Review

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

❖ Model-free

- 이미지 증강을 위해서 특별한 모델을 학습 시킬 필요 없는 경우
  - ➢ Single/Multiple-image : 이미지 1장 또는 여러 장을 활용하여 증강하는 경우
  - ➢ (Non)Instance-level : 이미지 자체를 직접 사용하여 증강하거나, 이미지 내의 객체를 추출하여 증강하는 경우

Table 2
Taxonomy with relevant methods.

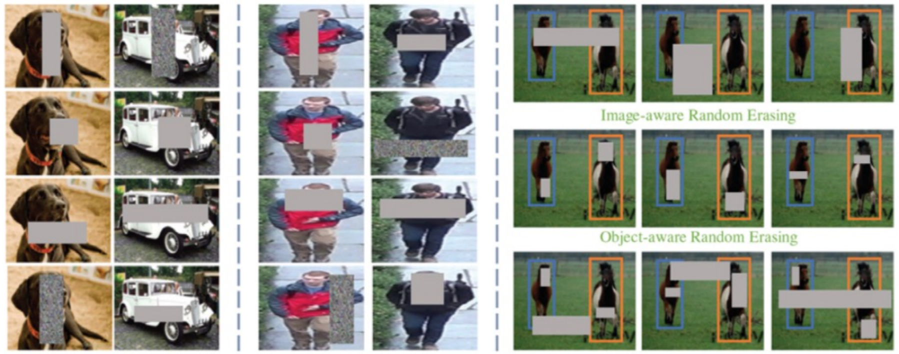
Categories			Relevant methods
Model-free	Single-image	Geometrical transformation	translation, rotation, flip, scale, elastic distortion.
		Color image processing	jittering.
		Intensity transformation	blurring and adding noise, Hide-and-Seek [23], Cutout [24], Random Erasing [25], GridMask [26].
	Multiple-image	Non-instance-level	SamplePairing [27], Mixup [28], BC Learning [29], CutMix [30], Mosaic [22], AugMix [31], PuzzleMix [32], Co-Mixup [33], SuperMix [34], GridMix [35].
		Instance-level	CutPas [36], Scale and Blend [37], Context DA [38], Simple CutPas [39], Continuous CutPas [40].
Model-based	Unconditional		DCGAN [41], [42–44]
	Label-conditional		BDA [45], ImbCGAN [46], BAGAN [47], DAGAN [48], MFC-GAN [49], IDA-GAN [50].
	Image-conditional	Label-preserving	S+U Learning [51], AugGAN [52], Plant-CGAN [53], StyleAug [54], Shape bias [55].
		Label-changing	EmoGAN [56], δ-encoder [57], Debiased NN [58], StyleMix [59], GAN-MBD [60], SCIT [2].
Optimizing policy-based	Reinforcement learning-based		AutoAugment [61], Fast AA [62], PBA [63], Faster AA [64], RandAugment [65], MADAO [66], LDA [67], LSSP [68].
	Adversarial learning-based		ADA [69], CDST-DA [70], AdaTransform [71], Adversarial AA [72], IF-DA [73], SPA [74].

Paper Review

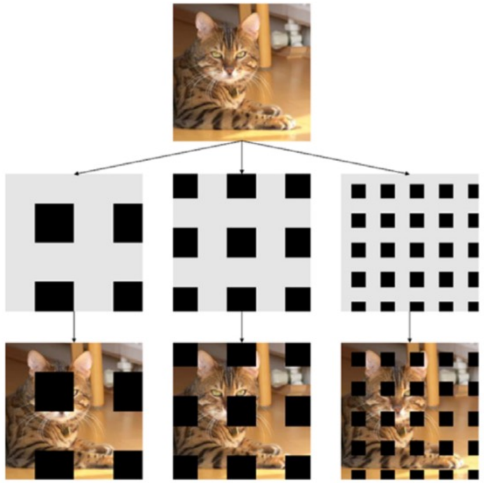
A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

❖ Model-free

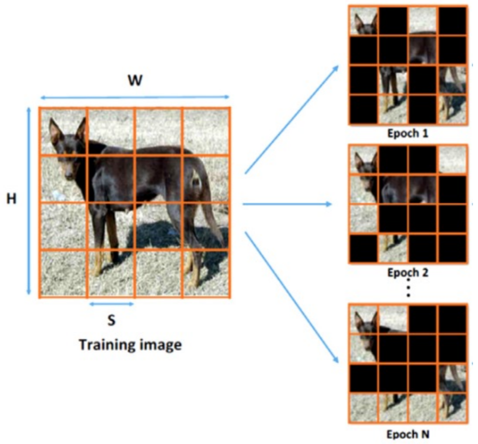
- 예제 : Random Erasing, Grid Mask, Hide-and-seek, …



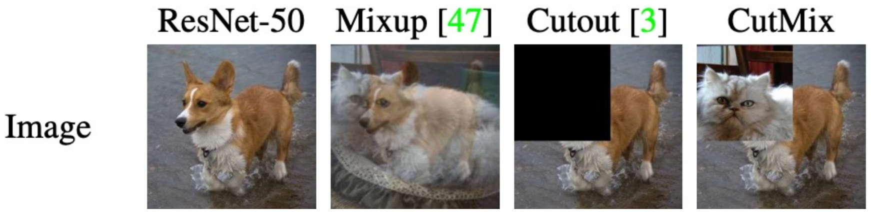
<Random Erasing>



<Grid Mask>



<Hide-and-seek>



<CutPas>

Zhong, Z., Zheng, L., Kang, G., Li, S., & Yang, Y. (2020, April). Random erasing data augmentation. In Proceedings of the AAAI conference on artificial intelligence (Vol. 34, No. 07, pp. 13001-13008).
Chen, P., Liu, S., Zhao, H., & Jia, J. (2020). Gridmask data augmentation. arXiv preprint arXiv:2001.04086.
Singh, K. K., & Lee, Y. J. (2017, October). Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In 2017 IEEE international conference on computer vision (ICCV) (pp. 3544-3553). IEEE.
Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 6023-6032).
Dwibedi, D., Misra, I., & Hebert, M. (2017). Cut, paste and learn: Surprisingly easy synthesis for instance detection. In Proceedings of the IEEE international conference on computer vision (pp. 1301-1310).

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

❖ Model-based

- 이미지 증강을 위해서 모델의 학습이 필요한 경우
  - ➢ Unconditional : 특별히 주어지는 조건 없이 이미지를 생성
  - ➢ Label/Image-conditional : Label이나 Image를 조건으로 주어져서 해당하는 이미지를 생성
  - ➢ Label-preserving/changing : Image가 주어져서 생성할 때 Label이 유지되거나/변화하거나

Table 2
Taxonomy with relevant methods.

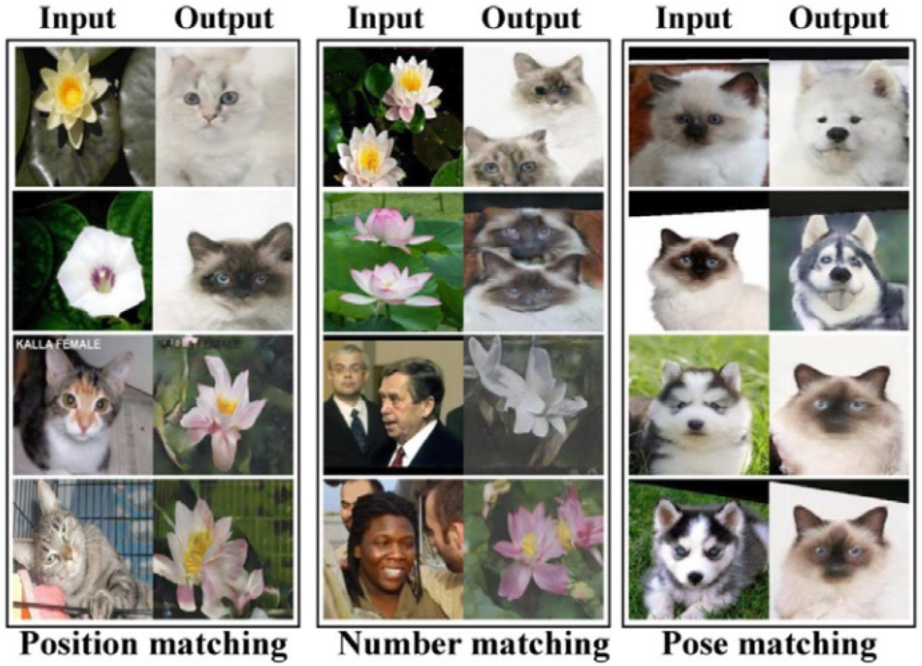
Categories			Relevant methods
Model-free	Single-image	Geometrical transformation	translation, rotation, flip, scale, elastic distortion.
		Color image processing	jittering.
		Intensity transformation	blurring and adding noise, Hide-and-Seek [23], Cutout [24], Random Erasing [25], GridMask [26].
	Multiple-image	Non-instance-level	SamplePairing [27], Mixup [28], BC Learning [29], CutMix [30], Mosaic [22], AugMix [31], PuzzleMix [32], Co-Mixup [33], SuperMix [34], GridMix [35].
		Instance-level	CutPas [36], Scale and Blend [37], Context DA [38], Simple CutPas [39], Continuous CutPas [40].
Model-based	Unconditional		DCGAN [41], [42–44]
	Label-conditional		BDA [45], ImbCGAN [46], BAGAN [47], DAGAN [48], MFC-GAN [49], IDA-GAN [50].
	Image-conditional	Label-preserving	S+U Learning [51], AugGAN [52], Plant-CGAN [53], StyleAug [54], Shape bias [55].
		Label-changing	EmoGAN [56], δ-encoder [57], Debiased NN [58], StyleMix [59], GAN-MBD [60], SCIT [2].
Optimizing policy-based	Reinforcement learning-based		AutoAugment [61], Fast AA [62], PBA [63], Faster AA [64], RandAugment [65], MADAO [66], LDA [67], LSSP [68].
	Adversarial learning-based		ADA [69], CDST-DA [70], AdaTransform [71], Adversarial AA [72], IF-DA [73], SPA [74].

Paper Review

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

❖ Model-based

- Image-conditional 예제 : GAN-MBD(Label-changing), StyleAug(Label-preserving)



<GAN-MBD>

<StyleAug>

Zheng, Z., Yu, Z., Wu, Y., Zheng, H., Zheng, B., & Lee, M. (2021). Generative adversarial network with multi-branch discriminator for imbalanced cross-species image-to-image translation. Neural Networks, 141, 355-371.
Jackson, P. T., Abarghouei, A. A., Bonner, S., Breckon, T. P., & Obara, B. (2019, June). Style augmentation: data augmentation via style randomization. In CVPR workshops (Vol. 6, pp. 10-11).

Paper Review

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

❖ Optimizing Policy-based

- 앞서 2가지 경우는 Domain Knowledge가 필요하다는 제약사항이 존재
  - ➢ Reinforcement Learning-based : 강화 학습을 활용하여 Domain Knowledge 없이 증강을 위한 최적 파라미터 도출
  - ➢ Adversarial Learning-based : 적대적인 관계(Large ↔ Small Loss)를 통해 최적 이미지 증강

Table 2
Taxonomy with relevant methods.

Categories			Relevant methods
Model-free	Single-image	Geometrical transformation	translation, rotation, flip, scale, elastic distortion.
		Color image processing	jittering.
		Intensity transformation	blurring and adding noise, Hide-and-Seek [23], Cutout [24], Random Erasing [25], GridMask [26].
	Multiple-image	Non-instance-level	SamplePairing [27], Mixup [28], BC Learning [29], CutMix [30], Mosaic [22], AugMix [31], PuzzleMix [32], Co-Mixup [33], SuperMix [34], GridMix [35].
		Instance-level	CutPas [36], Scale and Blend [37], Context DA [38], Simple CutPas [39], Continuous CutPas [40].
Model-based	Unconditional		DCGAN [41], [42–44]
	Label-conditional		BDA [45], ImbCGAN [46], BAGAN [47], DAGAN [48], MFC-GAN [49], IDA-GAN [50].
	Image-conditional	Label-preserving	S+U Learning [51], AugGAN [52], Plant-CGAN [53], StyleAug [54], Shape bias [55].
		Label-changing	EmoGAN [56], δ-encoder [57], Debiased NN [58], StyleMix [59], GAN-MBD [60], SCIT [2].
Optimizing policy-based	Reinforcement learning-based		AutoAugment [61], Fast AA [62], PBA [63], Faster AA [64], RandAugment [65], MADAO [66], LDA [67], LSSP [68].
	Adversarial learning-based		ADA [69], CDST-DA [70], AdaTransform [71], Adversarial AA [72], IF-DA [73], SPA [74].

Paper Review

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

❖ Reinforcement Learning-based

- 이미지를 직접적으로 생성하기 위하여 학습하는 것보다는, 이미지를 생성(증강)하는 방법을 학습하는 것(↔Model-based)
- 매우 큰 Search Space(이미지 증강 방법의 경우의 수)를 찾아야 하므로 Computational Cost가 매우 ↑
  - ➢ AutoAugment의 경우 $(16 \times 10 \times 11)^{2\times 5} \sim 2.9 \times 10^{32}$

Table 2
Taxonomy with relevant methods.

Categories			Relevant methods
Model-free	Single-image	Geometrical transformation	translation, rotation, flip, scale, elastic distortion.
		Color image processing	jittering.
		Intensity transformation	blurring and adding noise, Hide-and-Seek [23], Cutout [24], Random Erasing [25], GridMask [26].
	Multiple-image	Non-instance-level	SamplePairing [27], Mixup [28], BC Learning [29], CutMix [30], Mosaic [22], AugMix [31], PuzzleMix [32], Co-Mixup [33], SuperMix [34], GridMix [35].
		Instance-level	CutPas [36], Scale and Blend [37], Context DA [38], Simple CutPas [39], Continuous CutPas [40].
Model-based	Unconditional		DCGAN [41], [42–44]
	Label-conditional		BDA [45], ImbCGAN [46], BAGAN [47], DAGAN [48], MFC-GAN [49], IDA-GAN [50].
	Image-conditional	Label-preserving	S+U Learning [51], AugGAN [52], Plant-CGAN [53], StyleAug [54], Shape bias [55].
		Label-changing	EmoGAN [56], δ-encoder [57], Debiased NN [58], StyleMix [59], GAN-MBD [60], SCIT [2].
Optimizing policy-based	Reinforcement learning-based		AutoAugment [61], Fast AA [62], PBA [63], Faster AA [64], RandAugment [65], MADAO [66], LDA [67], LSSP [68].
	Adversarial learning-based		ADA [69], CDST-DA [70], AdaTransform [71], Adversarial AA [72], IF-DA [73], SPA [74].

Paper Review

Autoaugment: Learning augmentation strategies from data.

❖ Reinforcement Learning-based

- AutoAugment가 대표적인 방법론이며 많은 후속 연구가 진행
- 1816회 인용, 여러 방법론에서 성능 비교 대상으로 사용

AutoAugment:
Learning Augmentation Strategies from Data

Ekin D. Cubuk*†, Barret Zoph†, Dandelion Mané, Vijay Vasudevan, Quoc V. Le
Google Brain

Abstract

*Data augmentation is an effective technique for improving the accuracy of modern image classifiers. However, current data augmentation implementations are manually designed. In this paper, we describe a simple procedure called **AutoAugment** to automatically search for improved data augmentation policies. In our implementation, we have designed a search space where a policy consists of many sub-policies, one of which is randomly chosen for each image in each mini-batch. A sub-policy consists of two operations, each operation being an image processing function such as translation, rotation, or shearing, and the probabilities and magnitudes with which the functions are applied. We use a search algorithm to find the best policy such that the neural network yields the highest validation accuracy on a target dataset. Our method achieves state-of-the-art accuracy on CIFAR-10, CIFAR-100, SVHN, and ImageNet (without additional data). On ImageNet, we attain a Top-1 accuracy of 83.5% which is 0.4% better than the previous record of 83.1%. On CIFAR-10, we achieve an error rate of 1.5%, which is 0.6% better than the previous state-of-the-art. Augmentation policies we find are transferable between*

data domain: classifying an object is often insensitive to horizontal flips or translation. Network architectures can also be used to hardcode invariances: convolutional networks bake in translation invariance [16, 32, 25, 29]. However, using data augmentation to incorporate potential invariances can be easier than hardcoding invariances into the model architecture directly.

Dataset	GPU hours	Best published results	Our results
CIFAR-10	5000	2.1	1.5
CIFAR-100	0	12.2	10.7
SVHN	1000	1.3	1.0
Stanford Cars	0	5.9	5.2
ImageNet	15000	3.9	3.5

Table 1. Error rates (%) from this paper compared to the best results so far on five datasets (Top-5 for ImageNet, Top-1 for the others). Previous best result on Stanford Cars fine-tuned weights originally trained on a larger dataset [66], whereas we use a randomly initialized network. Previous best results on other datasets only include models that were not trained on additional data, for a single evaluation (without ensembling). See Tables 2,3, and 4 for more detailed comparison. GPU hours are estimated for an NVIDIA Tesla P100.

Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 113-123).

Paper Review

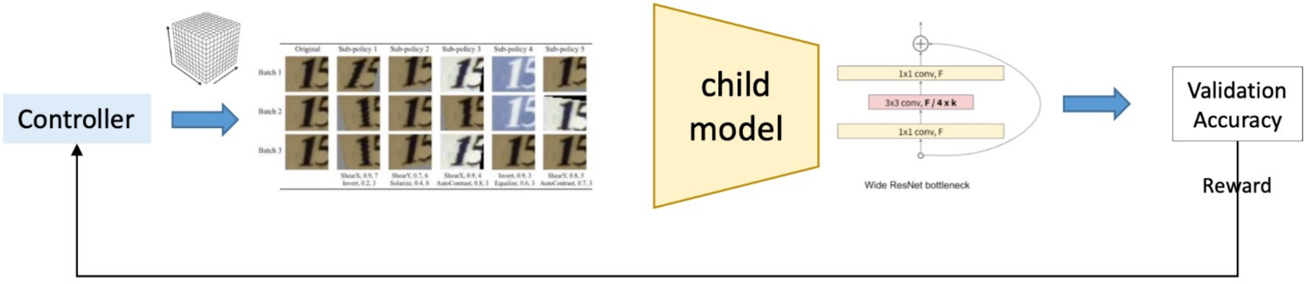
Autoaugment: Learning augmentation strategies from data.

❖ Reinforcement Learning-based

- 증강기법 적용 → Valid Set Accuracy 확인 → Controller를 통해 증강 기법 수정 → Valid Set Accuracy 확인 → (반복)

Operation Name	Description	Range of magnitudes
ShearX(Y)	Shear the image along the horizontal (vertical) axis with rate *magnitude*.	[-0.3,0.3]
TranslateX(Y)	Translate the image in the horizontal (vertical) direction by *magnitude* number of pixels.	[-150,150]
Rotate	Rotate the image *magnitude* degrees.	[-30,30]
AutoContrast	Maximize the the image contrast, by making the darkest pixel black and lightest pixel white.	
Invert	Invert the pixels of the image.	
Equalize	Equalize the image histogram.	
Solarize	Invert all pixels above a threshold value of *magnitude*.	[0,256]
Posterize	Reduce the number of bits for each pixel to *magnitude* bits.	[4,8]
Contrast	Control the contrast of the image. A *magnitude*=0 gives a gray image, whereas *magnitude*=1 gives the original image.	[0.1,1.9]
Color	Adjust the color balance of the image, in a manner similar to the controls on a colour TV set. A *magnitude*=0 gives a black & white image, whereas *magnitude*=1 gives the original image.	[0.1,1.9]
Brightness	Adjust the brightness of the image. A *magnitude*=0 gives a black image, whereas *magnitude*=1 gives the original image.	[0.1,1.9]
Sharpness	Adjust the sharpness of the image. A *magnitude*=0 gives a blurred image, whereas *magnitude*=1 gives the original image.	[0.1,1.9]
Cutout [12, 69]	Set a random square patch of side-length *magnitude* pixels to gray.	[0,60]
Sample Pairing [24, 68]	Linearly add the image with another image (selected at random from the same mini-batch) with weight *magnitude*, without changing the label.	[0, 0.4]

<List of Transformation>



<Architecture of AutoAugment>

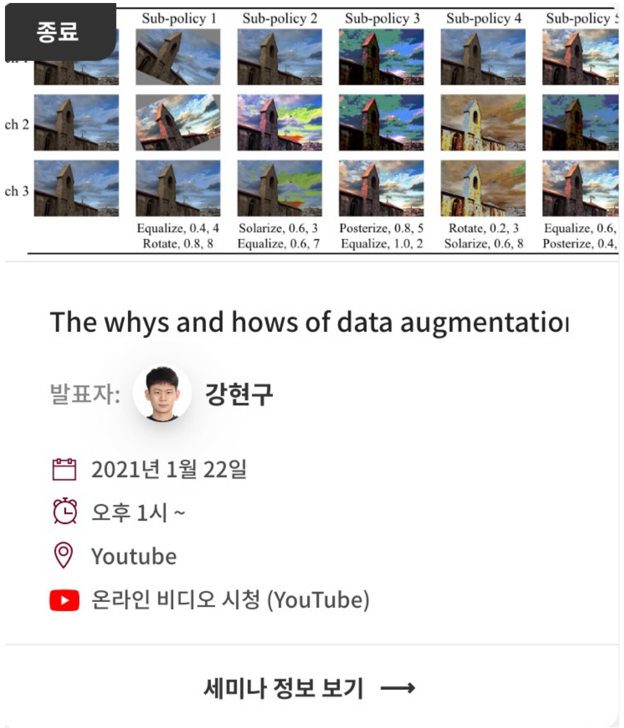
http://dmqa.korea.ac.kr/activity/seminar/370

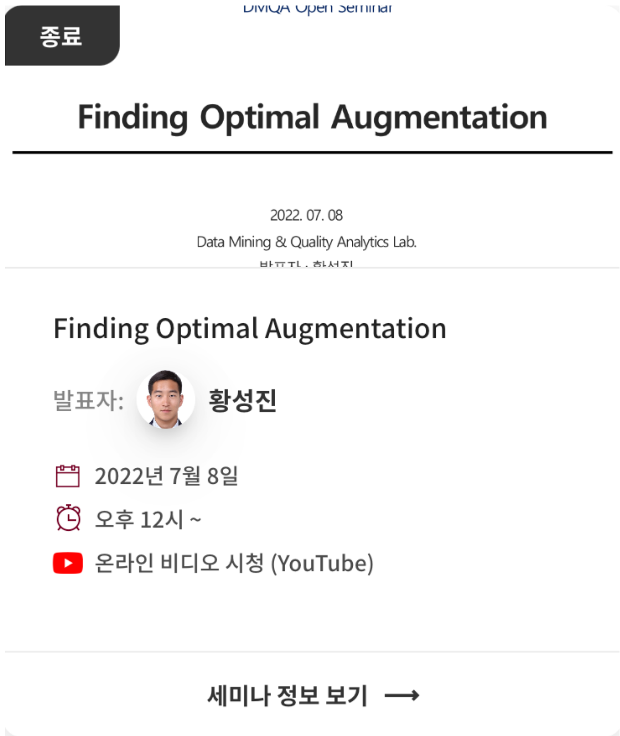
Paper Review

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

❖ Augmentation 관련 DMQA Open Seminar

- 다양한 Augmentation 소개 (Image, Text)
- AutoAugment와 후속 연구 흐름





http://dmqa.korea.ac.kr/activity/seminar/307, http://dmqa.korea.ac.kr/activity/seminar/370

Paper Review

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

❖ Adversarial Learning-based

- 가정 : ① Hard Sample은 모델을 일반화 하는데 더 유용, ② Large Training Loss를 만드는 Sample은 Hard Sample
  따라서, Large Training Loss를 만드는 Sample은 모델의 일반화에 도움을 준다.
- 목적 : 기존의 이미지를 Large Training Loss를 만드는 Hard Sample로 증강 (Loss를 줄이려는 일반적인 모델과 반대)

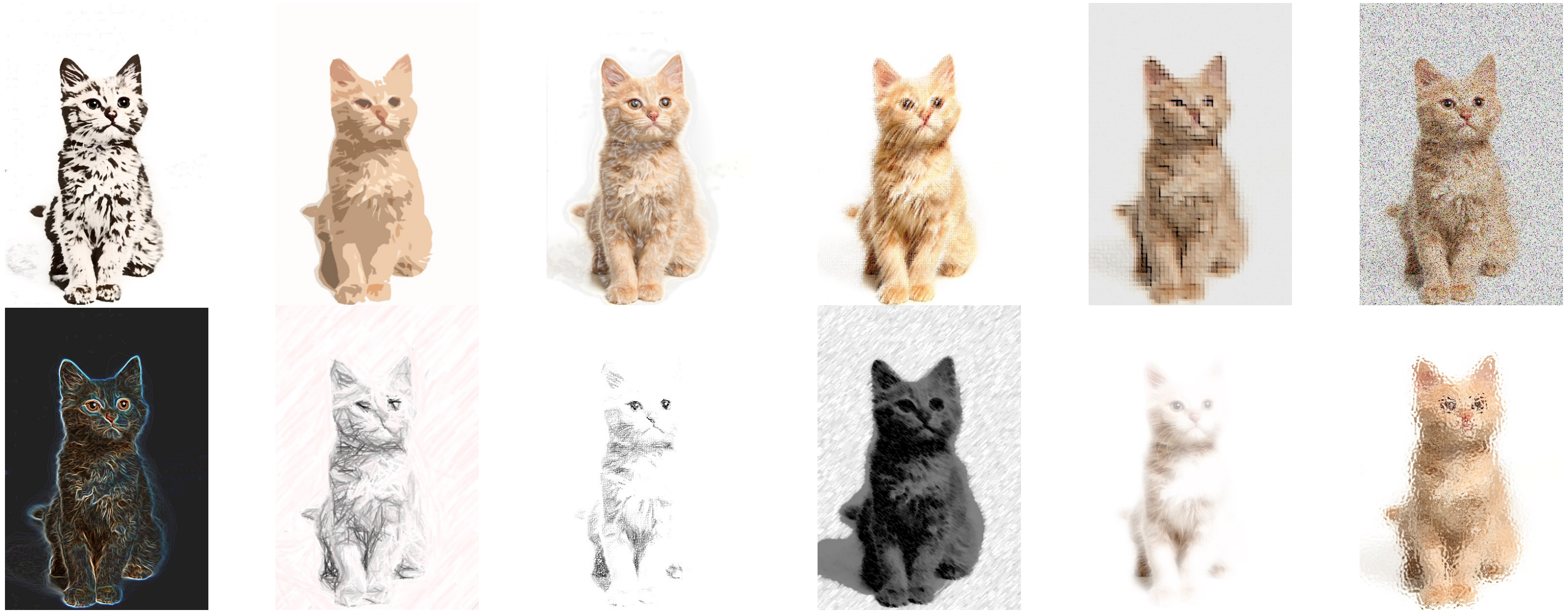


Image by master1305 on Freepik

Paper Review

A Comprehensive Survey of Image Augmentation Techniques for Deep Learning

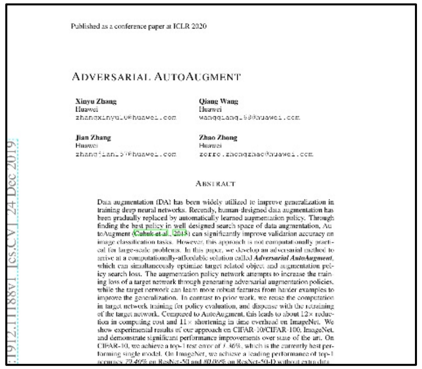
❖ Adversarial Learning-based

- ADA(2016), CDST-DA(2017), AdaTransform(2019), Adversarial AutoAugment(2020), IF-DA(2020), SPA(2021)

Reinforcement Learning Formula	
X	O
ADA, IF-DA, SPA	CDST-DA, AdaTransform, Adversarial AutoAugment

GAN Structure	
X	O
ADA, Adversarial AutoAugment, SPA	CDST-DA, AdaTransform, IF-DA

Augmentation Range	
Dataset	Sample
ADA, CDST-DA, AdaTransform, Adversarial AutoAugment, IF-DA	SPA

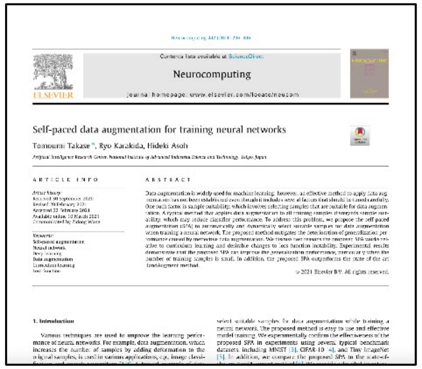

Published as a conference paper at ICLR 2020

ADVERSARIAL AUTOAUGMENT

Adversarial AutoAugment (2019)


Learning Augmentation Network via Influence Functions

Learning augmentation network via influence functions. (2020)


Neurocomputing

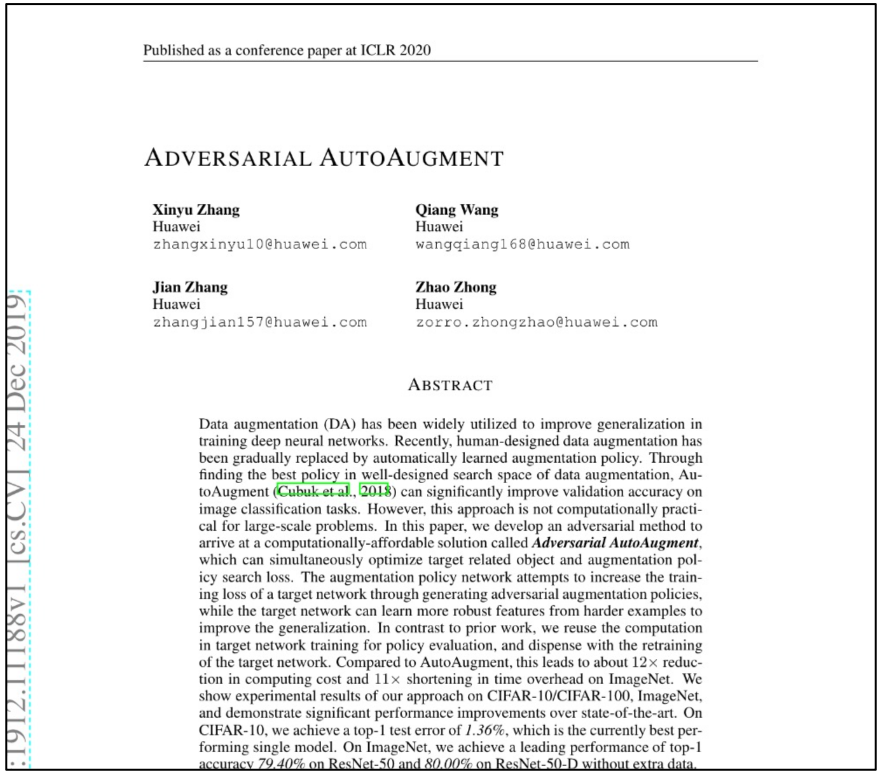
Self-paced data augmentation for training neural networks

Self-paced data augmentation for training neural networks (2021)

Paper Review
논문 리뷰

❖ Adversarial AutoAugment

- 2020년 ICLR에서 발표된 논문으로 현재까지 176회 인용
- Target Network, Policy Network를 동시에 학습하여 용인 가능한 수준의 계산 비용 달성



ADVERSARIAL AUTOAUGMENT

Xinyu Zhang
Huawei
zhangxinyu10@huawei.com

Qiang Wang
Huawei
wangqiang168@huawei.com

Jian Zhang
Huawei
zhangjian157@huawei.com

Zhao Zhong
Huawei
zorro.zhongzhao@huawei.com

ABSTRACT

Data augmentation (DA) has been widely utilized to improve generalization in training deep neural networks. Recently, human-designed data augmentation has been gradually replaced by automatically learned augmentation policy. Through finding the best policy in well-designed search space of data augmentation, AutoAugment (Cubuk et al., 2018) can significantly improve validation accuracy on image classification tasks. However, this approach is not computationally practical for large-scale problems. In this paper, we develop an adversarial method to arrive at a computationally-affordable solution called ***Adversarial AutoAugment***, which can simultaneously optimize target related object and augmentation policy search loss. The augmentation policy network attempts to increase the training loss of a target network through generating adversarial augmentation policies, while the target network can learn more robust features from harder examples to improve the generalization. In contrast to prior work, we reuse the computation in target network training for policy evaluation, and dispense with the retraining of the target network. Compared to AutoAugment, this leads to about 12× reduction in computing cost and 11× shortening in time overhead on ImageNet. We show experimental results of our approach on CIFAR-10/CIFAR-100, ImageNet, and demonstrate significant performance improvements over state-of-the-art. On CIFAR-10, we achieve a top-1 test error of *1.36%*, which is the currently best performing single model. On ImageNet, we achieve a leading performance of top-1 accuracy *79.40%* on ResNet-50 and *80.00%* on ResNet-50-D without extra data.

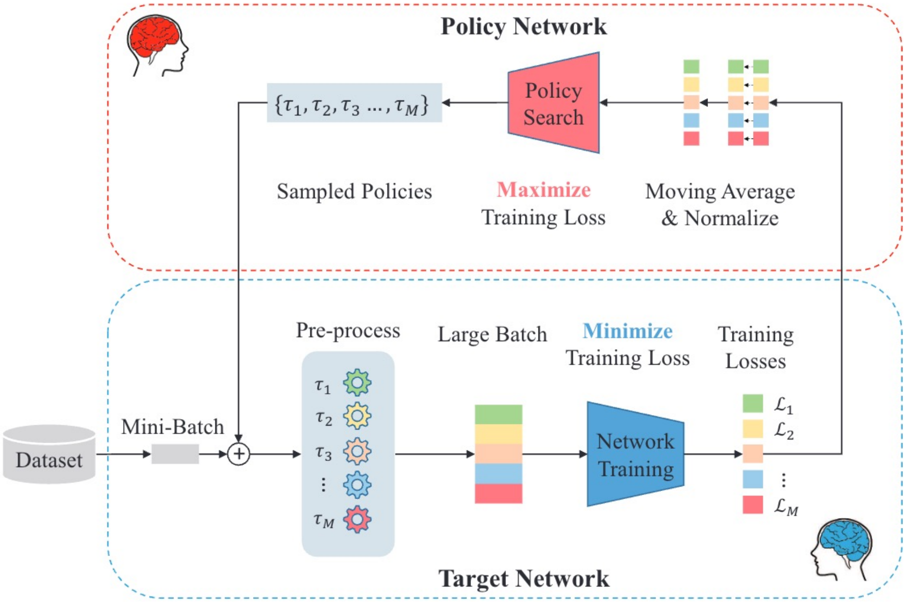
Zhang, X., Wang, Q., Zhang, J., & Zhong, Z. (2019). Adversarial autoaugment. arXiv preprint arXiv:1912.11188.

Paper Review

Adversarial AutoAugment

❖ 제안 방법론 특징

- 이미지 분류를 위한 Target Network와 최적 증강을 위한 Policy Network를 동시에 학습
- Target Network는 Loss를 최소화 하기 위하여 학습이 진행되며, Policy Network는 Loss를 최대화 하기 위하여 학습이 진행
- AutoAugment는 한번 Policy가 선택되면 1회 Training이 끝날 때까지 고정이지만, 본 방법론은 Batch마다 Policy가 변경됨
- REINFORCE 알고리즘 사용



Paper Review

Adversarial AutoAugment

❖ 제안 방법론

- Target Network $\mathcal{F}(\cdot, \boldsymbol{w})$, Policy Network $\mathcal{A}(\cdot, \boldsymbol{\theta})$, Loss Function $\mathcal{L}[\mathcal{F}(\boldsymbol{x}, \boldsymbol{w}), \boldsymbol{y}]$
- Random Data Augmentation $o(\cdot)$, Augmentation policy generated by Policy Network $\tau(\cdot)$

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\Omega} \mathcal{L}[\mathcal{F}(o(\boldsymbol{x}), \boldsymbol{w}), \boldsymbol{y}], \tag{1}$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \frac{1}{N} \sum_{n=1}^{N} \nabla_{\boldsymbol{w}} \mathcal{L}[\mathcal{F}(o(x_n), \boldsymbol{w}, y_n]. \tag{2}$$

Vanilla SGD

Random Aug이 아닌 Policy Network에 의한 Augmentation

$$\boldsymbol{w}^* = \arg\min_{\boldsymbol{w}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\Omega} \mathop{\mathbb{E}}_{\tau\sim\mathcal{A}(\cdot,\boldsymbol{\theta})} \mathcal{L}[\mathcal{F}(\tau(\boldsymbol{x}), \boldsymbol{w}), \boldsymbol{y}], \tag{3}$$

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \frac{1}{M \cdot N} \sum_{m=1}^{M} \sum_{n=1}^{N} \nabla_{\boldsymbol{w}} \mathcal{L}[\mathcal{F}(\tau_m(x_n), \boldsymbol{w}), y_n], \tag{4}$$

M개의 서로 다른 Policy $\{\tau_1, \tau_2, \cdots, \tau_M\}$

$$\mathcal{L}_m = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}[\mathcal{F}(\tau_m(x_n), \boldsymbol{w}), y_n]. \tag{5}$$

정의해주게 되면

$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \frac{1}{M} \sum_{m=1}^{M} \nabla_{\boldsymbol{w}} \mathcal{L}_m. \tag{6}$$

Target Netwrok 학습을 위한 식

Paper Review

Adversarial AutoAugment

❖ 제안 방법론

- Target Network $\mathcal{F}(\cdot, \boldsymbol{w})$, Policy Network $\mathcal{A}(\cdot, \boldsymbol{\theta})$, Loss Function $\mathcal{L}[\mathcal{F}(\boldsymbol{x}, \boldsymbol{w}), \boldsymbol{y}]$
- Random Data Augmentation $o(\cdot)$, Augmentation policy generated by Policy Network $\tau(\cdot)$

$$\begin{aligned} \boldsymbol{\theta}^* &= \arg\max_{\boldsymbol{\theta}} J(\boldsymbol{\theta}), \\ \text{where } J(\boldsymbol{\theta}) &= \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\Omega} \mathop{\mathbb{E}}_{\tau\sim\mathcal{A}(\cdot,\boldsymbol{\theta})} \mathcal{L}[\mathcal{F}(\tau(\boldsymbol{x}), \boldsymbol{w}), \boldsymbol{y}]. \end{aligned} \quad (7)$$

(3) 에선 $\arg min$ 이였다면 이젠 $\arg max$

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &= \nabla_{\boldsymbol{\theta}} \mathop{\mathbb{E}}_{\boldsymbol{x}\sim\Omega} \mathop{\mathbb{E}}_{\tau\sim\mathcal{A}(\cdot,\boldsymbol{\theta})} \mathcal{L}[\mathcal{F}(\tau(\boldsymbol{x}), \boldsymbol{w}), \boldsymbol{y}] \\ &\approx \sum_m \mathcal{L}_m \nabla_{\boldsymbol{\theta}} p_m = \sum_m \mathcal{L}_m p_m \nabla_{\boldsymbol{\theta}} \log p_m \\ &= \mathop{\mathbb{E}}_{\tau\sim\mathcal{A}(\cdot,\boldsymbol{\theta})} \mathcal{L}_m \nabla_{\boldsymbol{\theta}} \log p_m \\ &\approx \frac{1}{M} \sum_{m=1}^{M} \mathcal{L}_m \nabla_{\boldsymbol{\theta}} \log p_m, \end{aligned} \quad (8)$$

$$\begin{aligned} \nabla_{\boldsymbol{\theta}} J(\boldsymbol{\theta}) &\approx \frac{1}{M} \sum_{m=1}^{M} \widetilde{\mathcal{L}}_m \nabla_{\boldsymbol{\theta}} \log p_m, \\ \boldsymbol{\theta}_{e+1} &= \boldsymbol{\theta}_e + \beta \frac{1}{M} \sum_{m=1}^{M} \widetilde{\mathcal{L}}_m \nabla_{\boldsymbol{\theta}} \log p_m, \end{aligned} \quad (9)$$

분산을 줄이기 위해 Moving Average & Normalize

Policy Network 학습을 위한 식

❖ Policy Gradient Theorem

- Augmentation 작업은 미분 불가(non-differentiable)이기 때문에 Policy Network로의 Gradient Flow가 단절 됨
- 따라서 식 변형을 통해 충분한 수의 샘플로 구한 평균값으로 계산 해야함

$$\nabla \mathbb{E}_\pi[r(\tau)] = \nabla \int \pi(\tau)\, r(\tau)\, d\tau = \int \nabla \pi(\tau)\, r(\tau)\, d\tau$$

$$= \int \pi(\tau) \frac{\nabla \pi(\tau)}{\pi(\tau)}\, r(\tau)\, d\tau = \int \pi(\tau)\, r(\tau)\, \nabla \log \pi(\tau)\, d\tau$$

$$= \mathbb{E}_\pi[r(\tau)\, \nabla \log \pi(\tau)]$$

$$\text{and } \nabla \log \pi(\tau) \underset{\substack{\Uparrow \\ \text{Markov property} \\ p(s_{t+1} \mid s_t, a_t) = p(s_{t+1} \mid s_1, \ldots, a_t)}}{=} \nabla \log \left(p(s_0) \Pi_{t=0}^{T-1} \pi(a_t \mid s_t)\, p(s_{t+1} \mid s_t, a_t)\right) \underset{\substack{\Uparrow \\ \text{environment dynamic } p(s_{t+1} \mid s_t, a_t) \text{ is independent on } \theta \\ \nabla_\theta \log p(s_{t+1} \mid s_t, a_t) = 0}}{=} \sum \nabla \log \pi(a_t \mid s_t).$$

$$= \mathbb{E}_\pi[r(\tau) \sum \nabla \log \pi(a_t \mid s_t)]$$

❖ Policy Gradient Theorem

- Augmentation 작업은 미분 불가(non-differentiable)이기 때문에 Policy Network로의 Gradient Flow가 단절 됨
- 따라서 식 변형을 통해 충분한 수의 샘플로 구한 평균값으로 계산 해야함

$$\nabla \mathbb{E}_\pi[r(\tau)] = \nabla \int \pi(\tau)\, r(\tau)\, d\tau = \int \nabla \pi(\tau)\, r(\tau)\, d\tau$$

$$= \int \pi(\tau) \frac{\nabla \pi(\tau)}{\pi(\tau)}\, r(\tau)\, d\tau = \int \pi(\tau)\, r(\tau)\, \nabla \log \pi(\tau)\, d\tau$$

$$= \mathbb{E}_\pi[r(\tau)\, \nabla \log \pi(\tau)]$$

$$\text{and } \nabla \log \pi(\tau) \underset{\Uparrow}{=} \nabla \log \left(p(s_0) \Pi_{t=0}^{T-1} \pi(a_t \mid s_t)\, p(s_{t+1} \mid s_t, a_t)\right) \underset{\Uparrow}{=} \sum \nabla \log \pi(a$$

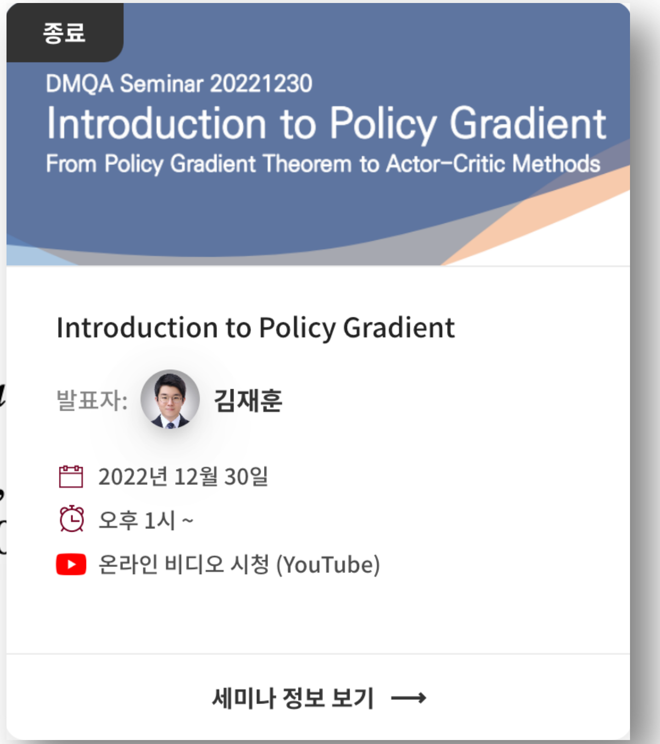
Markov property

$$p(s_{t+1} \mid s_t, a_t) = p(s_{t+1} \mid s_1, \ldots, a_t)$$

environment dynamic $p(s_{t+1} \mid s_t,$

$$\nabla_\theta \log p(s_{t+1} \mid s_t, a_t) = ($$

$$= \mathbb{E}_\pi[r(\tau) \sum \nabla \log \pi(a_t \mid s_t)]$$

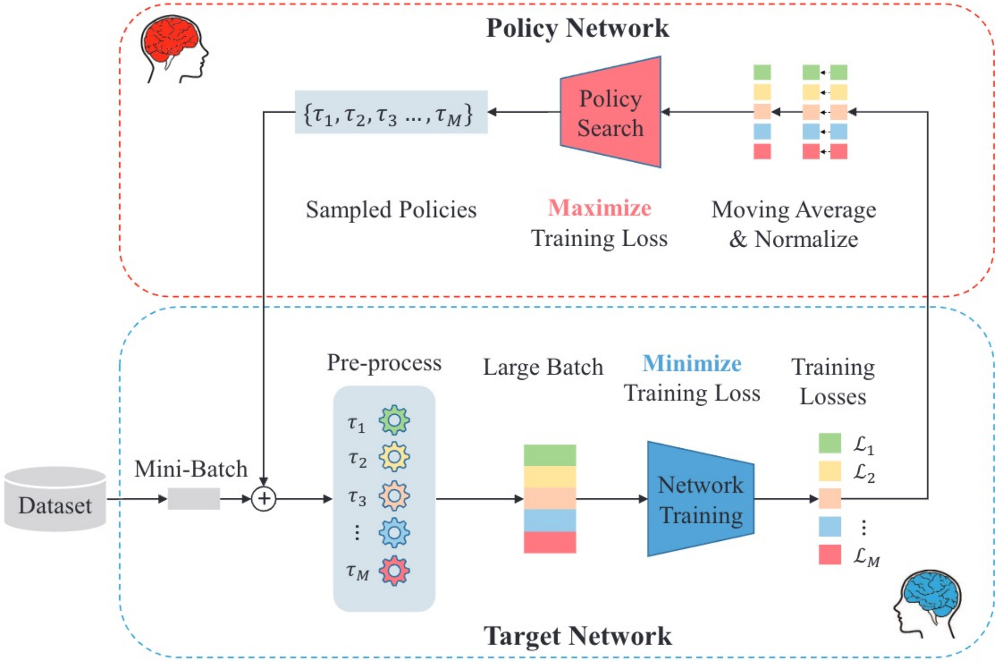


Paper Review

Adversarial AutoAugment

❖ 제안 방법론

$$\boldsymbol{\theta}_{e+1} = \boldsymbol{\theta}_e + \beta \frac{1}{M} \sum_{m=1}^{M} \widetilde{\mathcal{L}}_m \nabla_{\boldsymbol{\theta}} \log p_m,$$



$$\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta \frac{1}{M} \sum_{m=1}^{M} \nabla_{\boldsymbol{w}} \mathcal{L}_m.$$

Algorithm 1 Joint Training of Target Network and Augmentation Policy Network

Initialization: target network $\mathcal{F}(\cdot, \boldsymbol{w})$, augmentation policy network $\mathcal{A}(\cdot, \boldsymbol{\theta})$
Input: input examples $\boldsymbol{x}$, corresponding labels $\boldsymbol{y}$

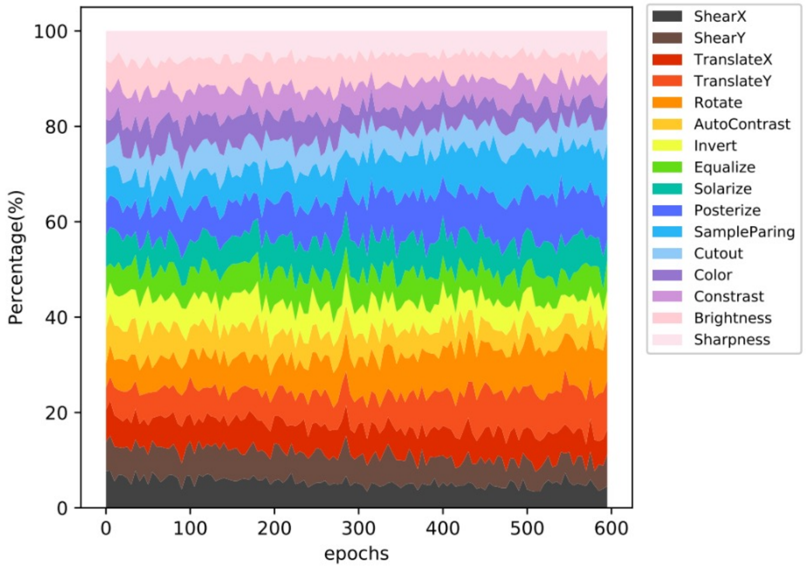
1: **for** $1 \le e \le epochs$ **do**
2: Initialize $\widehat{\mathcal{L}}_m = 0, \forall m \in \{1, 2, \cdots, M\}$;
3: Generate M policies with the probabilities $\{p_1, p_2, \cdots, p_M\}$;
4: **for** $1 \le t \le T$ **do**
5: Augment each batch data with M generated policies, respectively;
6: Update $\boldsymbol{w}_{e,t+1}$ according to Equation 4;
7: Update $\widehat{\mathcal{L}}_m$ through moving average, $\forall m \in \{1, 2, \cdots, M\}$;
8: Collect $\{\widehat{\mathcal{L}}_1, \widehat{\mathcal{L}}_2, \cdots, \widehat{\mathcal{L}}_M\}$;
9: Normalize $\widehat{\mathcal{L}}_m$ among M instances as $\widetilde{\mathcal{L}}_m, \forall m \in \{1, 2, \cdots, M\}$;
10: Update $\boldsymbol{\theta}_{e+1}$ via Equation 9;
11: Output $\boldsymbol{w}^*, \boldsymbol{\theta}^*$

Paper Review

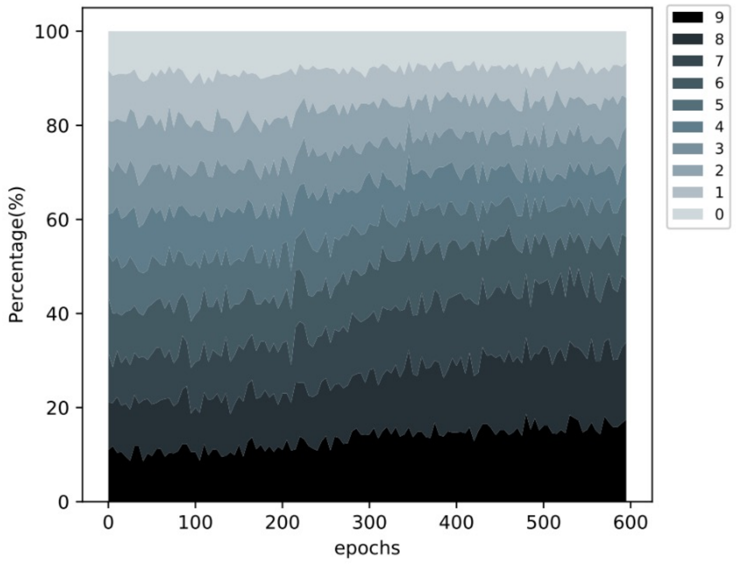
Adversarial AutoAugment

❖ 실험 결과

- 학습이 진행되면서 증강 Operation과 Magnitudes가 변하는 것을 시각적으로 확인 가능



(a) Operations



(b) Magnitudes

Paper Review

Adversarial AutoAugment

❖ 실험 결과

Table 1: Top-1 test error (%) on CIFAR-10. We replicate the results of Baseline, Cutout and AutoAugment methods from Cubuk et al. (2018), and the results of PBA from Ho et al. (2019) in all of our experiments.

Model	Baseline	Cutout	AutoAugment	PBA	Our Method
Wide-ResNet-28-10	3.87	3.08	2.68	2.58	**1.90±0.15**
Shake-Shake (26 2x32d)	3.55	3.02	2.47	2.54	**2.36±0.10**
Shake-Shake (26 2x96d)	2.86	2.56	1.99	2.03	**1.85±0.12**
Shake-Shake (26 2x112d)	2.82	2.57	1.89	2.03	**1.78±0.05**
PyramidNet+ShakeDrop	2.67	2.31	1.48	1.46	**1.36±0.06**

Table 2: Top-1 test error (%) on CIFAR-100.

Model	Baseline	Cutout	AutoAugment	PBA	Our Method
Wide-ResNet-28-10	18.80	18.41	17.09	16.73	**15.49±0.18**
Shake-Shake (26 2x96d)	17.05	16.00	14.28	15.31	**14.10±0.15**
PyramidNet+ShakeDrop	13.99	12.19	10.67	10.94	**10.42±0.20**

Table 3: Top-1 / Top-5 test error (%) on ImageNet. Note that the result of ResNet-50-D is achieved only through substituting the architecture.

Model	Baseline	AutoAugment	PBA	Our Method
ResNet-50	23.69 / 6.92	22.37 / 6.18	-	**20.60±0.15 / 5.53±0.05**
ResNet-50-D	22.84 / 6.48	-	-	**20.00±0.12 / 5.25±0.03**
ResNet-200	21.52 / 5.85	20.00 / 4.90	-	**18.68±0.18 / 4.70±0.05**

Paper Review

Adversarial AutoAugment

❖ 실험 결과

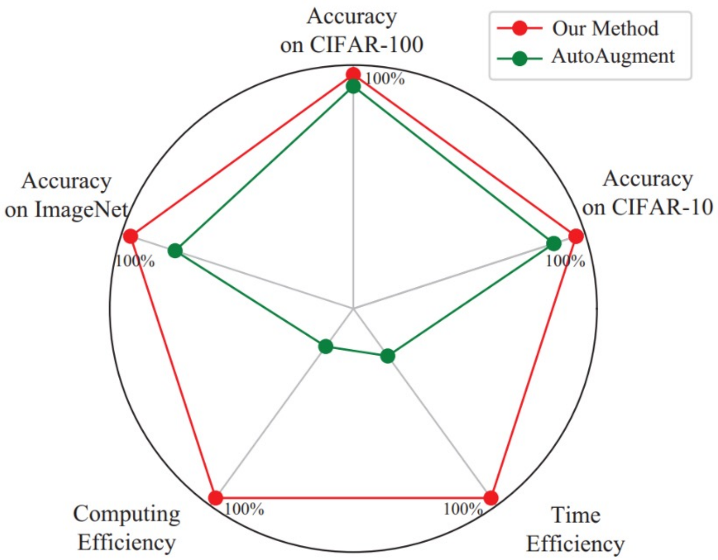


Figure 4: The Comparison of normalized performance between AutoAugment and our method. Please refer to the following tables for more details.

Table 5: The comparison of computing cost (GPU hours) and time overhead (days) in training ResNet-50 on ImageNet between AutoAugment and our method. The computing cost and time overhead are estimated on 64 NVIDIA Tesla V100s.

Method	Computing Cost			Time Overhead		
	Searching	Training	Total	Searching	Training	Total
AutoAugment	15000	160	15160	10	1	11
Our Method	∼0	1280	1280	∼0	1	1

Paper Review

논문 리뷰

❖ Learning augmentation network via influence functions (IF-DA)

- 2020년 CVPR에서 발표된 논문으로 현재까지 14회 인용
- Influence Function을 도입하여 반복된 재계산을 줄이고 미분 가능한 Augmentation 구조 제안

Learning Augmentation Network via Influence Functions

Donghoon Lee Hyunsin Park Trung Pham Chang D. Yoo
Korea Advanced Institute of Science and Technology (KAIST)
{iamdh, hs.park, trungpx, cd_yoo}@kaist.ac.kr

Abstract

Data augmentation can impact the generalization performance of an image classification model in a significant way. However, it is currently conducted on the basis of trial and error, and its impact on the generalization performance cannot be predicted during training. This paper considers an influence function that predicts how generalization performance, in terms of validation loss, is affected by a particular augmented training sample. The influence function provides an approximation of the change in validation loss without actually comparing the performances that include and exclude the sample in the training process. Based on this function, a differentiable augmentation network is learned to augment an input training sample to reduce validation loss. The augmented sample is fed into the classification network, and its influence is approximated as a function of the parameters of the last fully-connected layer of the classification network. By backpropagating the influence to the augmentation network, the augmentation network parameters are learned. Experimental results on CIFAR-10, CIFAR-100, and ImageNet show that the proposed method provides better generalization performance than conventional data augmentation methods do.

defined transformations, such as rotation, translation, cropping, scaling, and color perturbation, is a popular choice; however, choosing the transformations and determining the strength of each transformation, *e.g.* rotation angle, that result in the best performance is often conducted empirically through observations of validation loss [3, 33].

Most recently, strategies for composing the transformations through learning [6, 22, 28, 29, 35] rather than tuning by trial and error have been studied. To learn such a strategy, various learning criteria have been considered. Without considering a classification model, Ratner *et al.* [29] and Sixt *et al.* [35] adopted a strategy for augmenting realistic samples. Given a classification model, [22] and [28] consider antithetical strategies for augmenting samples that minimize and maximize training loss, respectively. It is not yet fully understood why these antithetical studies, which relate data augmentation to training loss, are effective in improving performance on test samples. Cubuk *et al.* [6] considered small child models to compute validation loss for augmenting samples to improve generalization performance. Here, learning requires a reinforcement learning framework with the validation loss as a reward. This necessitates the classification model to be learned from scratch for every update of the augmentation model parameters, requiring thousands

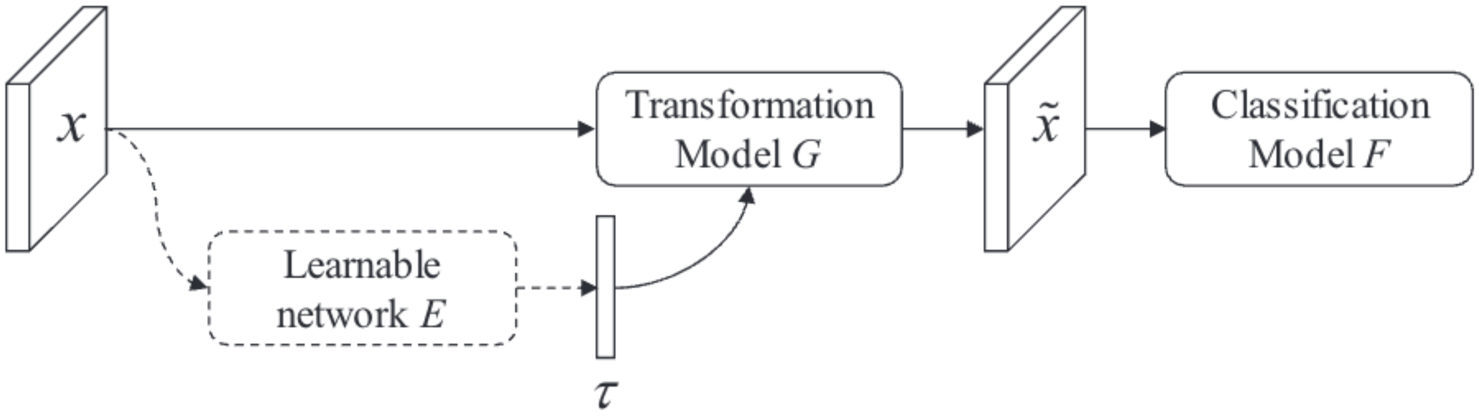
Lee, D., Park, H., Pham, T., & Yoo, C. D. (2020). Learning augmentation network via influence functions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10961-10970).

Paper Review

Learning augmentation network via influence functions

❖ 제안 방법론 특징

- Influence Function을 사용하여 Computational Cost를 줄이고 미분 가능한 Augmentation 구조 도입
- GAN(Generative Adversarial Networks) Framework에 기반하여 학습
- 이미지 증강은 Spatial/Appearance Transformation으로 구분
  - ➢ Spatial Transformation : Pixel 변환 (Flip, Crop, Scaling, Rotation, Shearing, Translation, Affine transformation)
  - ➢ Appearance Transformation : 색상 변환 (Contrast, Brightness, Color, Hue)



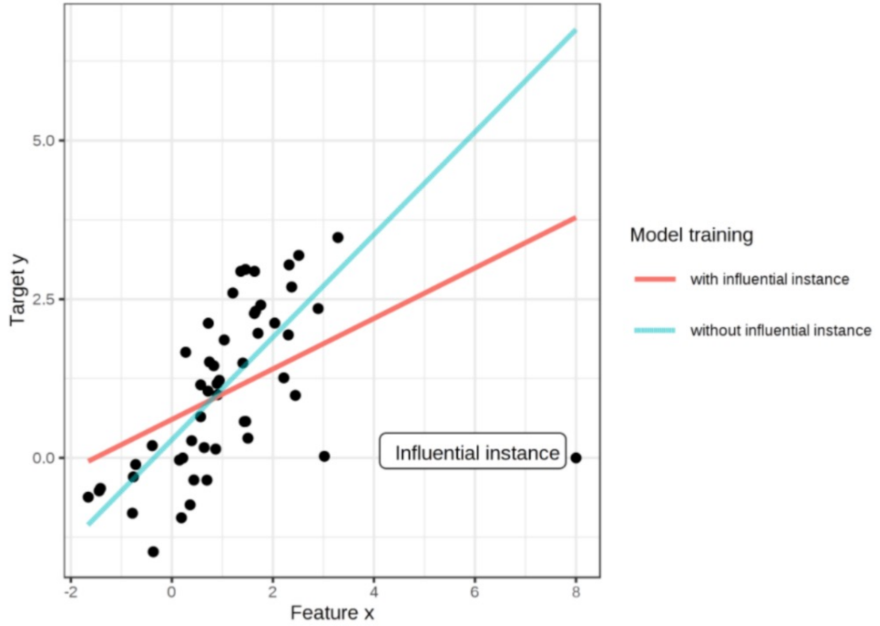
Paper Review

Learning augmentation network via influence functions

❖ Influential Instance

- 모델은 Train Dataset으로 학습된 결과물이기 때문에 Data Sample이 빠지거나 변화하면 모델에 영향을 줌
- 특정 Sample의 제거가 모델의 Parameter 또는 Prediction에 큰 영향을 주게 되면 모델을 해석하는데 도움을 줄 수 있다.

  즉, 이러한 Sample을 **<u>Influential Instance</u>**라 함



https://christophm.github.io/interpretable-ml-book/
https://hellopotatoworld.tistory.com/13
https://www.youtube.com/watch?v=xlmlY8WHjkU&ab_channel=TerryTaeWoongUm

Paper Review

Learning augmentation network via influence functions

❖ How to find Influential Instance?

- DFBETA

$$DFBETA_i = \beta - \beta^{(-i)}$$

- β : 모든 Data를 사용해 학습된 모델의 weight 벡터
- β(-i) : i번째 Instance를 제거하고 재학습된 모델의 weight 벡터

- Cook's Distance

$$D_i = \frac{\sum_{j=1}^{n}(\hat{y}_j - \hat{y}_j^{(-i)})^2}{p \cdot MSE}$$

- 분자 : i번째 Instance를 포함했을 때와 안 했을 때의 prediction의 Squared Difference
- 분모 : Feature의 개수 p와 전체 MSE의 곱 (모든 Instance에 대하여 동일)

https://christophm.github.io/interpretable-ml-book/
https://hellopotatoworld.tistory.com/13
https://www.youtube.com/watch?v=xlmlY8WHjkU&ab_channel=TerryTaeWoongUm

❖ How to find Influential Instance?

- DFBETA

$$DFBETA_i = \beta - \beta^{(-i)}$$

- β : 모든 Data를 사용해 학습된 모델의 weight 벡터
- β(-i) : i번째 Instance를 제거하고 재학습된 모델의 weight 벡터

- Cook's Distance

$$D_i = \frac{\sum_{j=1}^{n}(\hat{y}_j - \hat{y}_j^{(-i)})^2}{p \cdot MSE}$$

- 분자 : i번째 Instance를 포함했을 때와 안 했을 때의 prediction의 Squared Difference
- 분모 : Feature의 개수 p와 전체 MSE의 곱 (모든 Instance에 대하여 동일)

Instance를 삭제할 때마다 모델을 재학습 해야 하는 비효율 발생

https://christophm.github.io/interpretable-ml-book/
https://hellopotatoworld.tistory.com/13
https://www.youtube.com/watch?v=xlmlY8WHjkU&ab_channel=TerryTaeWoongUm

Paper Review

Learning augmentation network via influence functions

❖ Influence Function

- 모델을 매번 재학습할 필요 없이, Training Loss를 통해 Instance들의 가중치를 변경하며 모델이 얼마나 변화할지 근사

Optimal model param. : $\hat{\theta} \stackrel{\text{def}}{=} \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta)$

Model param. by training w/o z : $\hat{\theta}_{-z} \stackrel{\text{def}}{=} \arg\min_{\theta \in \Theta} \sum_{z_i \neq z} L(z_i, \theta)$

Model param. by upweighting z : $\hat{\theta}_{\epsilon,z} \stackrel{\text{def}}{=} \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L(z_i, \theta) + \epsilon L(z, \theta)$

without z == ($\epsilon = -\frac{1}{n}$)

- The influence of upweighting z on the parameters θ

$$\mathcal{I}_{\text{up,params}}(z) \stackrel{\text{def}}{=} \left.\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta})$$

- The influence of upweighting z on the loss at a test point

$$\begin{aligned}\mathcal{I}_{\text{up,loss}}(z, z_{\text{test}}) &\stackrel{\text{def}}{=} \left.\frac{dL(z_{\text{test}}, \hat{\theta}_{\epsilon,z})}{d\epsilon}\right|_{\epsilon=0} \\ &= \nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^{\top} \left.\frac{d\hat{\theta}_{\epsilon,z}}{d\epsilon}\right|_{\epsilon=0} \\ &= -\nabla_{\theta} L(z_{\text{test}}, \hat{\theta})^{\top} H_{\hat{\theta}}^{-1} \nabla_{\theta} L(z, \hat{\theta}).\end{aligned}$$

https://christophm.github.io/interpretable-ml-book/
https://hellopotatoworld.tistory.com/13
https://www.youtube.com/watch?v=xlmlY8WHjkU&ab_channel=TerryTaeWoongUm

Paper Review

Learning augmentation network via influence functions

❖ Influence Function

$$\mathcal{I}_{\text{up, params}}(z_i) \triangleq \left.\frac{d\hat{\theta}(\mathbf{z}^{\text{tr}} \cup \epsilon z_i)}{d\epsilon}\right|_{\epsilon=0} \tag{8}$$

$$= -H(\hat{\theta}(\mathbf{z}^{\text{tr}}))^{-1}\nabla_\theta l(z_i, \hat{\theta}(\mathbf{z}^{\text{tr}})). \tag{9}$$

$$\mathcal{I}_{\text{up, loss}}(z_i, z_j) \triangleq \left.\frac{dl(z_j, \hat{\theta}(\mathbf{z}^{\text{tr}} \cup \epsilon z_i))}{d\epsilon}\right|_{\epsilon=0} \tag{11}$$

$$= \nabla_\theta l(z_j, \hat{\theta}(\mathbf{z}^{\text{tr}}))^\top \left.\frac{d\hat{\theta}(\mathbf{z}^{\text{tr}} \cup \epsilon z_i)}{d\epsilon}\right|_{\epsilon=0} \tag{12}$$

$$= -\nabla_\theta l(z_j, \hat{\theta}(\mathbf{z}^{\text{tr}}))^\top H(\hat{\theta}(\mathbf{z}^{\text{tr}}))^{-1}\nabla_\theta l(z_i, \hat{\theta}(\mathbf{z}^{\text{tr}})). \tag{13}$$

$$\mathcal{I}_{\text{up, loss}}(z_i, \mathbf{z}^{\text{val}})$$
$$= -\nabla_\theta \mathcal{L}(\mathbf{z}^{\text{val}}, \hat{\theta}(\mathbf{z}^{\text{tr}}))^\top H(\hat{\theta}(\mathbf{z}^{\text{tr}}))^{-1}\nabla_\theta l(z_i, \hat{\theta}(\mathbf{z}^{\text{tr}})). \tag{14}$$

https://christophm.github.io/interpretable-ml-book/
https://hellopotatoworld.tistory.com/13
https://www.youtube.com/watch?v=xlmlY8WHjkU&ab_channel=TerryTaeWoongUm

Learning augmentation network via influence functions

❖ Influence Function

- $\mathcal{I}_{\text{aug, loss}}(z_i, \tilde{z}_i, \mathbf{z}^{\text{val}})$ 를 계산함으로써 Augmented 된 Sample과 Original Sample의 Influence Function 값이 차이를 계산 가능
- 즉, Loss를 최대화 시키는 방향으로 Original Sample인 z_i을 변형 시켜서 Augmented Sample인 $\tilde{z}_i$를 구할 수 있다.

$$
\begin{aligned}
&\mathcal{I}_{\text{aug, loss}}(z_i, \tilde{z}_i, \mathbf{z}^{\text{val}}) \\
&\quad \triangleq \left. \frac{d\mathcal{L}(\mathbf{z}^{\text{val}}, \hat{\theta}(\mathbf{z}^{\text{tr}} \cup \epsilon\tilde{z}_i \backslash \epsilon z_i))}{d\epsilon} \right|_{\epsilon=0} && (16) \\
&\quad = \nabla_\theta \mathcal{L}(\mathbf{z}^{\text{val}}, \hat{\theta}(\mathbf{z}^{\text{tr}}))^\top \left. \frac{d\hat{\theta}(\mathbf{z}^{\text{tr}} \cup \epsilon\tilde{z}_i \backslash \epsilon z_i)}{d\epsilon} \right|_{\epsilon=0} && (17) \\
&\quad = -\nabla_\theta \mathcal{L}(\mathbf{z}^{\text{val}}, \hat{\theta}(\mathbf{z}^{\text{tr}}))^\top H(\hat{\theta}(\mathbf{z}^{\text{tr}}))^{-1} \\
&\qquad \left(\nabla_\theta l(\tilde{z}_i, \hat{\theta}(\mathbf{z}^{\text{tr}})) - \nabla_\theta l(z_i, \hat{\theta}(\mathbf{z}^{\text{tr}}))\right) && (18) \\
&\quad = \mathcal{I}_{\text{up, loss}}(\tilde{z}_i, \mathbf{z}^{\text{val}}) - \mathcal{I}_{\text{up, loss}}(z_i, \mathbf{z}^{\text{val}}). && (19)
\end{aligned}
$$

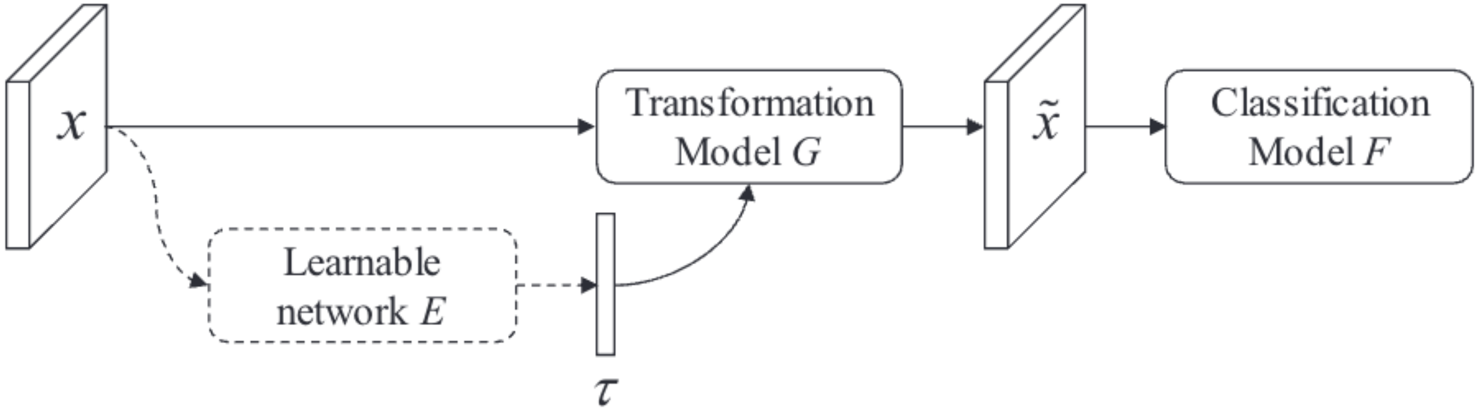
https://christophm.github.io/interpretable-ml-book/
https://hellopotatoworld.tistory.com/13
https://www.youtube.com/watch?v=xlmlY8WHjkU&ab_channel=TerryTaeWoongUm

Paper Review

Learning augmentation network via influence functions

❖ 제안 방법론

- GAN Framework로 G를 사전학습
- iHVPS (inverse Hessian-vector products) 계산
- Influence가 최대가 되도록 E & G 학습
- F 네트워크 재학습

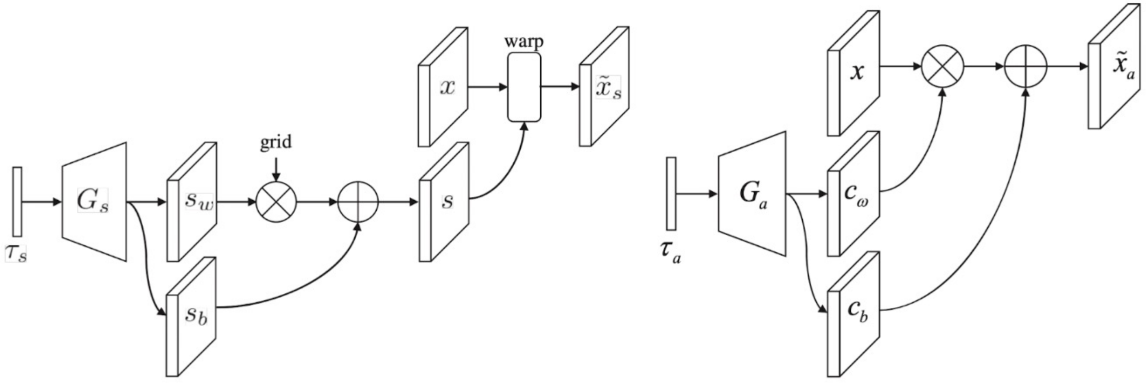


Paper Review

Learning augmentation network via influence functions

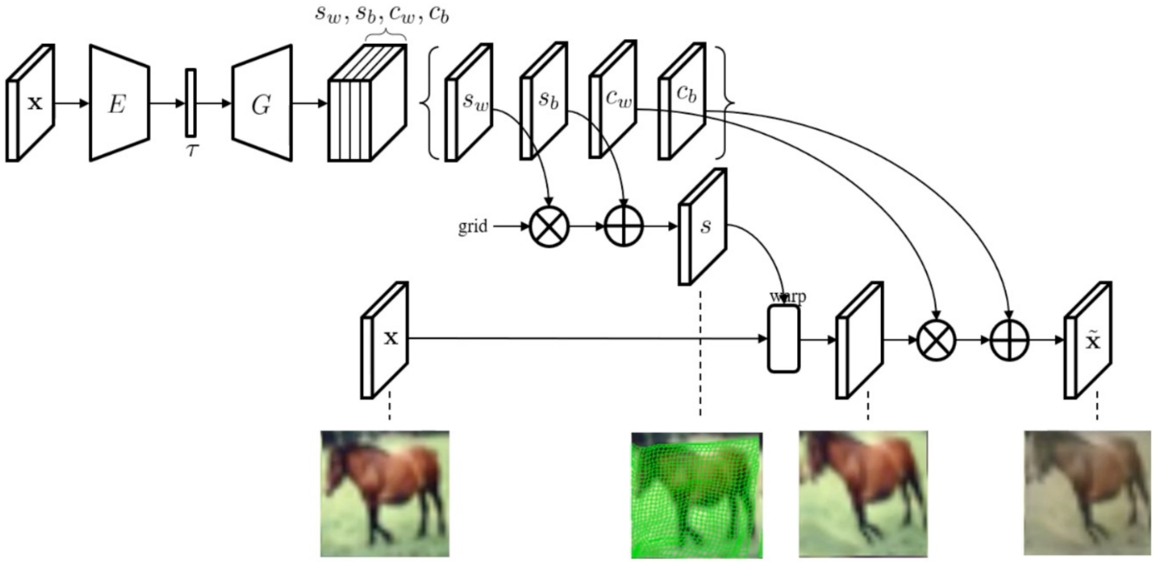
❖ Spatial/Appearance Transformation

- 좌표이동이 수반되는 Spatial Transformation은 weight의 곱, bias의 합 그리고 Bilinear Interpolation에 의하여 x를 $\tilde{x}$로 변형
- Bilinear Interpolation은 미분 가능
- Appearance Transformation은 $x + \delta x$의 형태로 좌표 이동이 없는 Spatial Transformation과 동일



(a) Spatial transformation model

(b) Appearance transformation model



Paper Review

Learning augmentation network via influence functions

❖ 실험 결과

- AutoAugment보다 성능은 약간 낮지만 속도는 600배 더 빠르다는 장점

Dataset	%	Model	None	Heur.	Ratner MF [29]	Ratner LSTM [29]	Proposed
MNIST	1	4 layer CNN	9.8	4.1	3.5	3.3	3.1
	10	4 layer CNN	2.7	1.0	0.8	0.9	0.8
CIFAR-10	10	ResNet-56 [15]	34.0	22.5	20.2	18.5	17.7
CIFAR-10	100	ResNet-56 [15]	12.2	7.7	5.6	6.0	5.2
CIFAR-100	100	ResNet-56 [15]	36.3	31.6	-	-	29.6

Model	Baseline [6]	Baseline (ours)	AutoAug. [6]	Proposed
ResNet-50 [15]	76.3 / 93.1	76.1 / 93.0	77.6 / 93.8	77.1 / 93.4
ResNet-200 [15]	78.5 / 94.2	78.1 / 94.0	80.0 / 95.0	79.0 / 94.6

Table 3: Validation set Top-1 / Top-5 accuracy (%) on ImageNet dataset. The experiments are conducted under the same setting as [6]. All results are obtained using 1-crop testing.

Dataset	AutoAug. [6]	Proposed
CIFAR-10	5,000	8
ImageNet	15,000	40

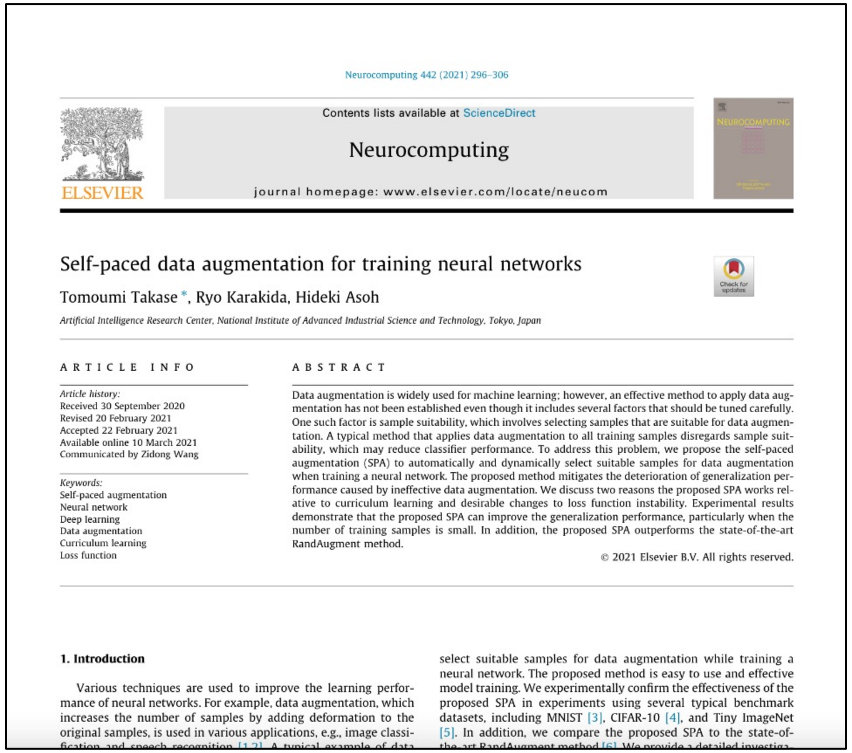
Table 4: GPU hours comparison of AutoAugment and the proposed method. Ours are estimated with Titan-X Pascal.

Paper Review

논문 리뷰

❖ Self-paced data augmentation for training neural networks

- 2021년 Neurocomputing에서 발표된 논문으로 현재까지 12회 인용
- Train Dataset 전부 증강을 적용하던 다른 방법론과 달리, 증강에 적합한 이미지만 증강하는 방법 제안



Neurocomputing 442 (2021) 296–306

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Self-paced data augmentation for training neural networks

Tomoumi Takase *, Ryo Karakida, Hideki Asoh

Artificial Intelligence Research Center, National Institute of Advanced Industrial Science and Technology, Tokyo, Japan

ARTICLE INFO

Article history:
Received 30 September 2020
Revised 20 February 2021
Accepted 22 February 2021
Available online 10 March 2021
Communicated by Zidong Wang

Keywords:
Self-paced augmentation
Neural network
Deep learning
Data augmentation
Curriculum learning
Loss function

ABSTRACT

Data augmentation is widely used for machine learning; however, an effective method to apply data augmentation has not been established even though it includes several factors that should be tuned carefully. One such factor is sample suitability, which involves selecting samples that are suitable for data augmentation. A typical method that applies data augmentation to all training samples disregards sample suitability, which may reduce classifier performance. To address this problem, we propose the self-paced augmentation (SPA) to automatically and dynamically select suitable samples for data augmentation when training a neural network. The proposed method mitigates the deterioration of generalization performance caused by ineffective data augmentation. We discuss two reasons the proposed SPA works relative to curriculum learning and desirable changes to loss function instability. Experimental results demonstrate that the proposed SPA can improve the generalization performance, particularly when the number of training samples is small. In addition, the proposed SPA outperforms the state-of-the-art RandAugment method.

1. Introduction

Various techniques are used to improve the learning performance of neural networks. For example, data augmentation, which increases the number of samples by adding deformation to the original samples, is used in various applications, e.g., image classi-

select suitable samples for data augmentation while training a neural network. The proposed method is easy to use and effective model training. We experimentally confirm the effectiveness of the proposed SPA in experiments using several typical benchmark datasets, including MNIST [3], CIFAR-10 [4], and Tiny ImageNet [5]. In addition, we compare the proposed SPA to the state-of-

Takase, T., Karakida, R., & Asoh, H. (2021). Self-paced data augmentation for training neural networks. Neurocomputing, 442, 296-306.

Paper Review

Self-paced data augmentation for training neural networks

❖ 제안 방법론 특징

- 모든 이미지에 증강을 적용하는 것보다 증강에 적합한 Sample들만 증강하는 것이 효율적
- Curriculum Learning과 Loss Function Instability에 초점을 맞추어 설계

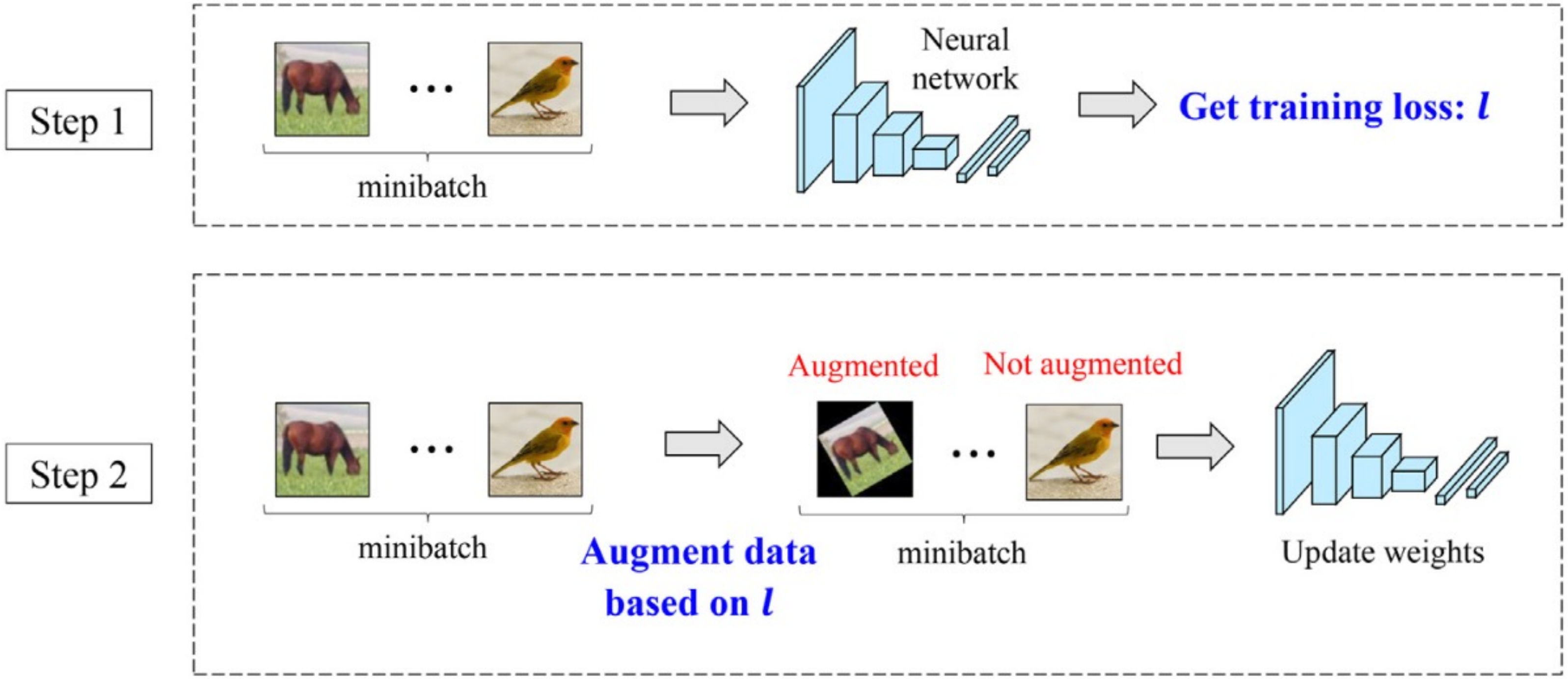
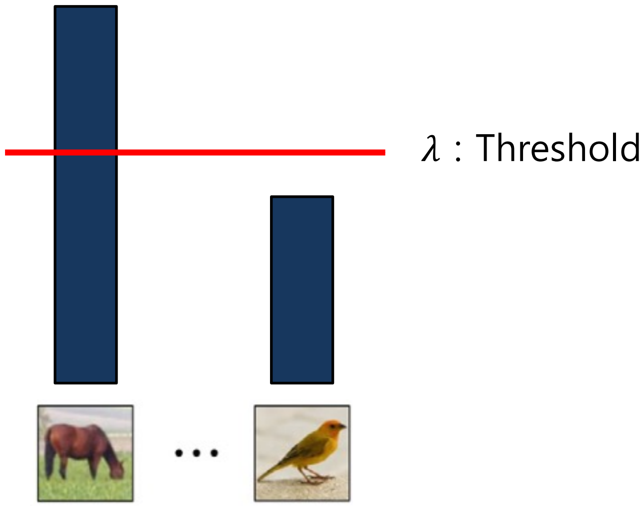


Fig. 2. SPA training procedure. Steps 1 and 2 are performed for each minibatch.



Paper Review

Self-paced data augmentation for training neural networks

❖ 제안 방법론

- Curriculum Learning : 쉬운 Sample부터 학습하여 어려운 Sample을 학습하도록 난이도를 고려한 학습 순서 설정

  (학습초기 Loss가 클 때 증강이 많이 되며 점차 줄어드는 구조이므로 Original한 의미와는 반대)

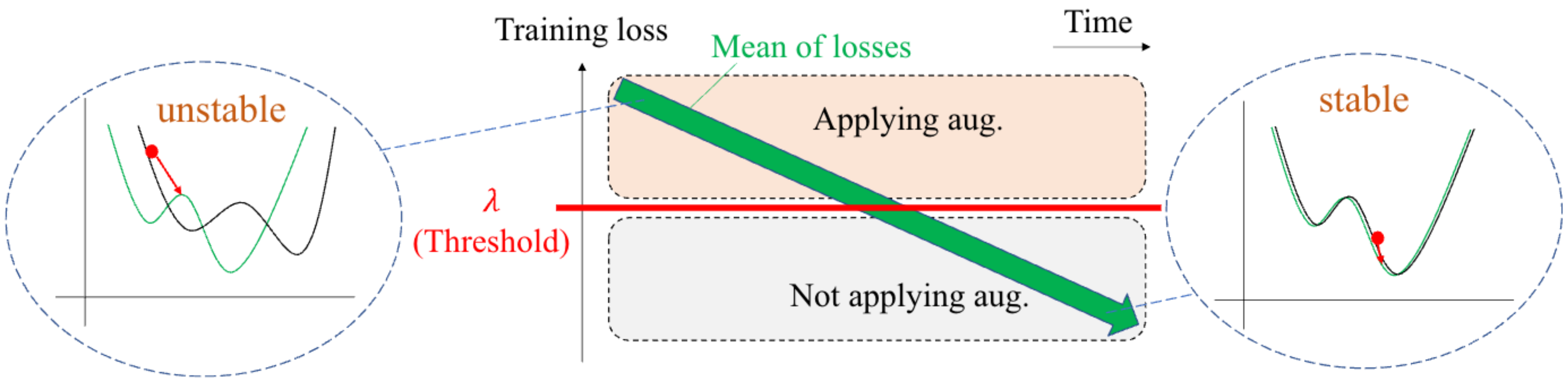


Fig. 3. Criterion by which data augmentation is applied in SPA. The concept of curriculum learning and desirable change of loss function instability is summarized. "Aug." denotes data augmentation.

Paper Review

Self-paced data augmentation for training neural networks

❖ 제안 방법론

- Loss Function Instability : 데이터 증강을 하면 Loss의 Variance 및 크기는 커짐 (Fig. 5)

  보다 안정적인 Loss를 유도하여 Local Minima에 빠지지 않도록 방지

  증강을 하지 않았을 때보다 일반화된 모델을 생성하면서, Loss의 변화는 최소화

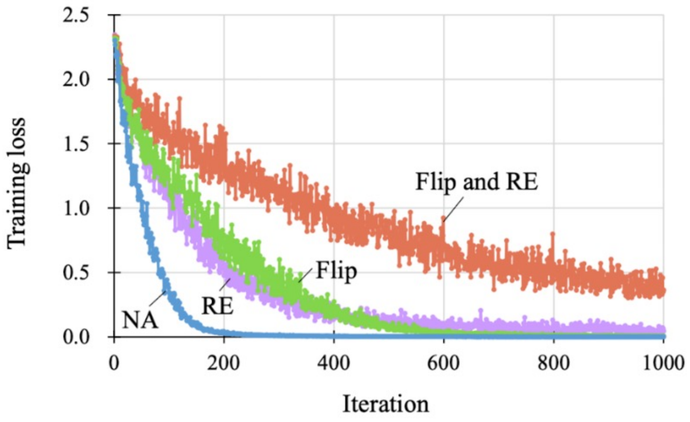
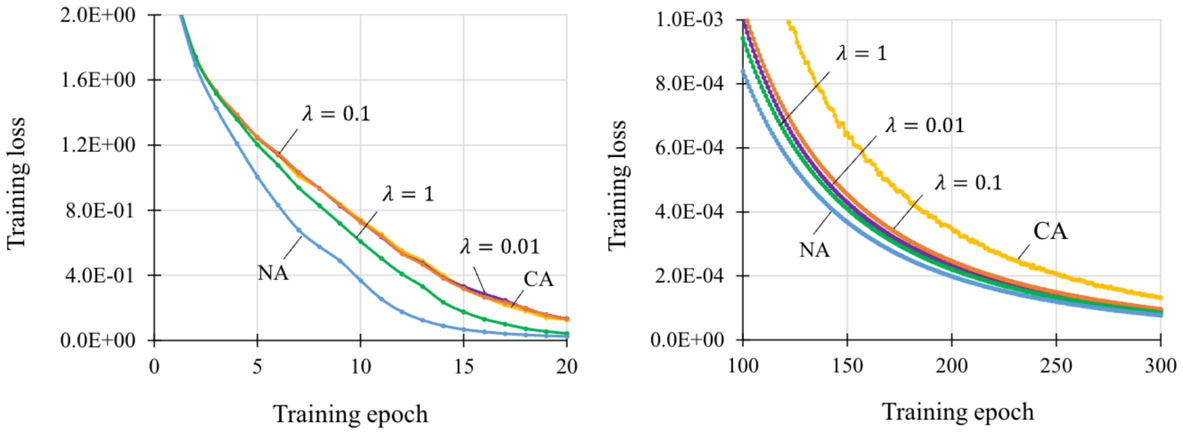


Fig. 5. Effects of data augmentation on training loss instability. RE denotes random erasing. In cases other than NA, data augmentation was applied to all training samples.



Paper Review

Self-paced data augmentation for training neural networks

❖ 실험 결과

Table 1

Best test accuracy (%) when training the small CNN using several number of training samples. Values in parentheses are the standard error. Numbers in bold indicate the highest accuracy, and those in underline indicate considerably greater values than other methods. A data augmentation is regarded as being unsuitable when the accuracy for CA is considerably smaller than that for NA. "Number" denotes the number of training samples, and "trans." denotes the translation method.

Dataset	Number	Aug.	SPA ($\lambda = 0.1$)	CA ($\lambda = 0$)	NA ($\lambda = \infty$)	Unsuitable
CIFAR-10	100	flip	**30.87 (±0.33)**	30.84 (±0.25)	28.52 (±0.56)	
		translation	32.28 (±0.48)	**33.00 (±0.48)**	28.52 (±0.56)	
	500	flip	**45.09 (±0.21)**	43.84 (±0.30)	42.23 (±0.36)	
		translation	48.29 (±0.59)	**49.11 (±0.75)**	42.23 (±0.36)	
	1000	flip	**52.78 (±0.38)**	50.19 (±0.35)	48.47 (±0.49)	
		translation	**56.96 (±0.42)**	56.16 (±0.28)	48.47 (±0.49)	
	5000	flip	**66.44 (±0.41)**	63.51 (±0.39)	64.01 (±0.19)	
		translation	**72.80 (±0.18)**	70.35 (±0.13)	64.01 (±0.19)	
	10,000	flip	**69.86 (±0.14)**	68.10 (±0.09)	69.28 (±0.35)	✔
		translation	**75.98 (±0.11)**	75.08 (±0.15)	69.28 (±0.35)	
	(all) 50,000	flip	77.71 (±0.13)	77.70 (±0.12)	**78.32 (±0.11)**	✔
		translation	83.02 (±0.07)	**83.11 (±0.18)**	78.32 (±0.11)	
MNIST	100	flip	**85.71 (±0.65)**	74.25 (±0.90)	84.12 (±0.92)	✔
		translation	**92.14 (±0.18)**	90.02 (±0.42)	84.12 (±0.92)	
	500	flip	**96.81 (±0.11)**	89.82 (±0.14)	96.69 (±0.02)	✔
		translation	**97.72 (±0.05)**	97.20 (±0.07)	96.69 (±0.02)	
	1000	flip	**97.77 (±0.04)**	93.36 (±0.02)	**97.77 (±0.10)**	✔
		translation	**98.52 (±0.02)**	98.23 (±0.04)	97.77 (±0.10)	
	5000	flip	**98.88 (±0.05)**	96.91 (±0.05)	98.80 (±0.04)	✔
		translation	**99.15 (±0.02)**	99.00 (±0.01)	98.80 (±0.04)	
	10,000	flip	**99.27 (±0.01)**	97.86 (±0.03)	99.18 (±0.04)	✔
		translation	**99.40 (±0.02)**	99.21 (±0.01)	99.18 (±0.04)	
	(all) 60,000	flip	**99.58 (±0.02)**	98.76 (±0.06)	**99.58 (±0.04)**	✔
		translation	99.60 (±0.01)	**99.62 (±0.01)**	99.58 (±0.04)	

Paper Review

Self-paced data augmentation for training neural networks

❖ 실험 결과

- 데이터셋이 작을 경우에 효과를 보이며, Augmentation을 1개씩만 적용하였다는 한계점

Table 3
Best test accuracy (%) when training WideResNet28-10 using all samples from CIFAR-10, Fashion-MNIST, and SVHN datasets. We evaluated SPA using several λ values, CA, and NA. The results for NA have been added under each dataset in Dataset column. Values in parentheses after accuracies are the standard error. Numbers in bold indicate the highest accuracy, and underlined values indicate considerably greater values than those of CA and NA.

Dataset	Augmentation	SPA ($\lambda = 0.01$)	SPA ($\lambda = 0.1$)	SPA ($\lambda = 1$)	CA ($\lambda = 0$)
CIFAR-10 (NA: 92.43 (±0.11))	mixup [3]	<u>**92.66**</u> (±<u>**0.07**</u>)	92.48 (±0.09)	92.65 (±0.09)	92.24 (±0.03)
	cutout [4]	**94.01** (±**0.10**)	93.05 (±0.06)	92.62 (±0.07)	93.99 (±0.06)
	random erasing [5]	**92.45** (±**0.08**)	92.38 (±0.06)	92.38 (±0.06)	92.30 (±0.03)
	RICAP [6]	95.94 (±0.04)	86.47 (±0.54)	68.19 (±2.13)	<u>**96.29**</u> (±<u>**0.07**</u>)
	flip and crop	<u>**95.05**</u> (±<u>**0.05**</u>)	<u>**95.05**</u> (±<u>**0.07**</u>)	93.03 (±0.08)	90.91 (±0.27)
	translation and rotation	91.37 (±0.09)	<u>**93.79**</u> (±<u>**0.04**</u>)	92.04 (±0.13)	23.55 (±0.87)
Fashion-MNIST (NA: 93.85 (±0.07))	mixup [3]	94.39 (±0.05)	94.39 (±0.03)	94.44 (±0.06)	**94.45** (±**0.07**)
	cutout [4]	94.49 (±0.04)	94.19 (±0.04)	94.23 (±0.05)	<u>**94.64**</u> (±<u>**0.04**</u>)
	random erasing [5]	**93.90** (±**0.04**)	93.89 (±0.03)	93.87 (±0.07)	93.88 (±0.05)
	RICAP [6]	94.81 (±0.08)	94.72 (±0.03)	91.37 (±0.37)	<u>**95.06**</u> (±<u>**0.04**</u>)
	flip and crop	94.09 (±0.03)	<u>**94.62**</u> (±<u>**0.04**</u>)	93.93 (±0.08)	87.76 (±0.18)
	translation and rotation	93.57 (±0.04)	93.02 (±0.08)	**93.88** (±**0.05**)	**93.88** (±**0.04**)
SVHN (NA: 96.45 (±0.02))	mixup [3]	96.50 (±0.03)	96.57 (±0.03)	<u>**97.14**</u> (±<u>**0.04**</u>)	96.43 (±0.03)
	cutout [4]	<u>**97.05**</u> (±<u>**0.02**</u>)	96.63 (±0.02)	96.86 (±0.02)	96.24 (±0.03)
	random erasing [5]	96.54 (±0.04)	96.54 (±0.04)	96.47 (±0.04)	**96.55** (±**0.03**)
	RICAP [6]	97.33 (±0.03)	93.87 (±0.19)	70.49 (±2.80)	<u>**97.47**</u> (±<u>**0.04**</u>)
	flip and crop	96.68 (±0.04)	97.05 (±0.05)	<u>**97.07**</u> (±<u>**0.06**</u>)	95.27 (±0.06)
	translation and rotation	96.61 (±0.04)	97.06 (±0.04)	<u>**97.27**</u> (±<u>**0.03**</u>)	88.50 (±0.55)

Conclusion

Conclusion

❖ 결론

- Image augmentation?
- Taxonomy
- Paper Reviews
  - ➢ Adversarial AutoAugment (+Reinforcement Learning Formula)
  - ➢ IF-DA (+Influence Function)
  - ➢ SPA

Reference

❖ Papers

- Xu, M., Yoon, S., Fuentes, A., & Park, D. S. (2023). A comprehensive survey of image augmentation techniques for deep learning. Pattern Recognition, 109347.
- Zhang, X., Wang, Q., Zhang, J., & Zhong, Z. (2019). Adversarial autoaugment. arXiv preprint arXiv:1912.11188.
- Lee, D., Park, H., Pham, T., & Yoo, C. D. (2020). Learning augmentation network via influence functions. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 10961-10970).
- Takase, T., Karakida, R., & Asoh, H. (2021). Self-paced data augmentation for training neural networks. Neurocomputing, 442, 296-306.
- Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., & Le, Q. V. (2019). Autoaugment: Learning augmentation strategies from data. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 113-123).
- Fawzi, A., Samulowitz, H., Turaga, D., & Frossard, P. (2016, September). Adaptive data augmentation for image classification. In 2016 IEEE international conference on image processing (ICIP) (pp. 3688-3692). Ieee.
- Ratner, A. J., Ehrenberg, H., Hussain, Z., Dunnmon, J., & Ré, C. (2017). Learning to compose domain-specific transformations for data augmentation. Advances in neural information processing systems, 30.
- Tang, Z., Peng, X., Li, T., Zhu, Y., & Metaxas, D. N. (2019). Adatransform: Adaptive data transformation. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 2998-3006).

❖ YouTube

- https://www.youtube.com/watch?v=xlmlY8WHjkU&ab_channel=TerryTaeWoongUm
- http://dmqa.korea.ac.kr/activity/seminar/307
- http://dmqa.korea.ac.kr/activity/seminar/370

❖ Website

- https://christophm.github.io/interpretable-ml-book/
- https://hellopotatoworld.tistory.com/13

고맙습니다