

# RandAugment: Practical automated data augmentation with a reduced search space

Ekin D. Cubuk \* , Barret Zoph\* , Jonathon Shlens, Quoc V. Le

Google Research, Brain Team

{cubuk, barretzoph, shlens, qvl}@google.com

- Introduction
- Related work
- RandAugment
- Systematic failures of a separate proxy task
- Results
  - CIFAR&SVHN results
  - ImageNet results
  - COCO results
  - Additional results
- Discussion

- 데이터 증강(Data Augmentation)은 데이터 기반의 기계학습 알고리즘 성능을 향상시키기 위해서 흔하게 사용되고 있는 방법 중 하나
- 데이터를 증강시키기 위해서는 **도메인에 대한 사전 지식과 수작업을** 필요로 함
  - 다른 애플리케이션이나 도메인에 확장하여 사용하기 어렵게 만드는 단점을 가짐
- 최적의 신경망의 구조를 학습을 통해서 찾는 NAS(Network Architecture Search)와 같이 데이터 증강에 필요한 **증강 규칙**을 학습을 통해 **자동으로 설계하는 방법들이 등장**
  - PBA(Population Based Augment), AutoAugment, Fast AutoAugment
  - 전통적인 데이터 증강 방법들의 단점을 해결할 수 있는 잠재력을 보여줌

- 학습된 데이터 증강 방법은 이미지 분류를 위한 준지도 학습의 정확도와 견고성을 크게 개선했으며, MS-COCO와 PASCAL-VOC를 이용한 객체 탐지 분야에서도 성능을 개선
- 학습을 통해 설계된 데이터 증강 규칙의 이점에도 불구하고, 두 개의 개별 최적화 절차는 엄청난 규모의 추가적인 연산 뿐만 아니라 최적화 절차의 복잡도를 증가시킴
- PBA(Population Based Augment), Fast AutoAugment 는 더 효과적인 검색 방법에 대해서 발견했으나, 여전히 기계 학습 모델을 학습하는데 있어 추가적인 계산 비용과 복잡도를 증가시키는 분리된 최적화 절차가 필요함

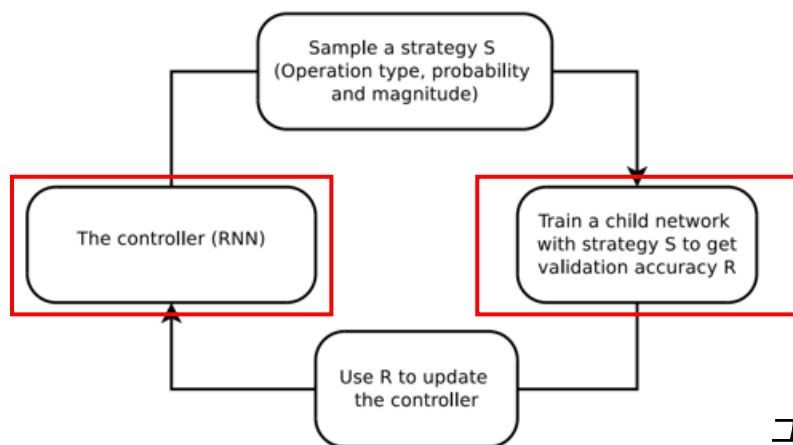


그림. 1 AutoAugment의 데이터 증강 규칙 학습 방법

<https://arxiv.org/pdf/1805.09501.pdf>

<https://pod3275.github.io/paper/2019/07/13/FastAutoAugment.html>

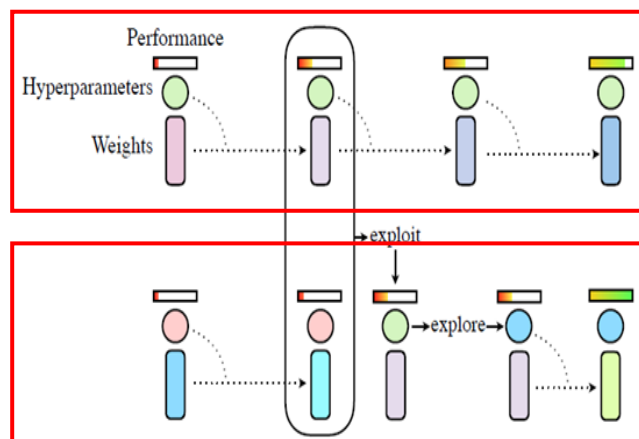


그림. 2 Population Based Augmentation의 데이터 증강 규칙 학습 방법

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

표. 1 데이터셋에 따른 가장 높은 성능을 기록한 실험의 결과, AutoAugment로 학습한 모델의 정확도와 학습시간

- 기존의 학습을 통한 데이터 증강 방법들은 small proxy task에서 학습한 결과가 large target task에서도 최적의 결과를 여준다 그대로 적용될 수 있다는 것을 가정
  - proxy task가 large task에 대한 증강 규칙에 대한 예측 결과를 제공하는 것에 대한 가정과 같음
  - 전체 데이터의 일부를 통해 학습한 증강 규칙을 전체 데이터의 증강 규칙으로 사용했을 때 높은 성능을 보여줌 (PBA)
- 본 논문의 저자들은 small proxy task에서 학습한 결과가 large target task에서도 최적의 결과를 보여준다는 가정에 대해 실험적 결과를 통해 반박
  - 특히, 데이터 증강의 강도가 모델과 데이터셋의 크기에 크게 의존하기 때문에 proxy task의 결과가 최선이 아닌 차선이라는 것을 보임
  - proxy task에서 분리된 검색 과정을 제거할 수 있는 경우 데이터 증강 성능이 개선될 수 있음을 시사

- 데이터 증강은 심층 신경망을 이용한 컴퓨터비전 분야의 모델 학습에 중요한 역할을 수행
  - 자연 영상에서 horizontal flips, random cropping과 translations는 분류와 객체 탐지에서 일반적으로 사용
- MNIST에서 elastic distortions across scale, position, and orientation은 좋은 결과를 보임
  - 데이터 증강 방법은 다양하며, **데이터 도메인에 따라서 데이터 증강 전략이 달라져야 함**

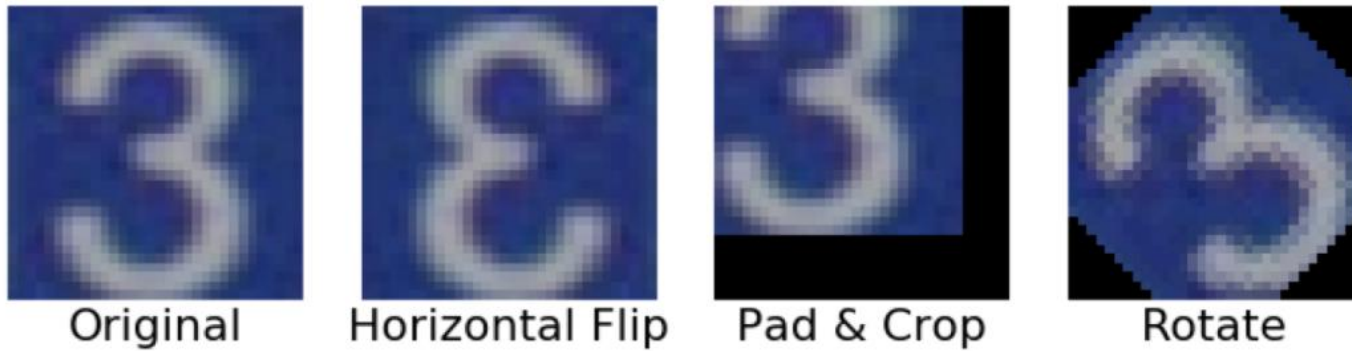


그림. 1 숫자 3이 표현된 원본 그림(Original)과 기본적인 증강 방법이 적용된 그림(Horizontal Flip, Pad and Crop, Rotate)

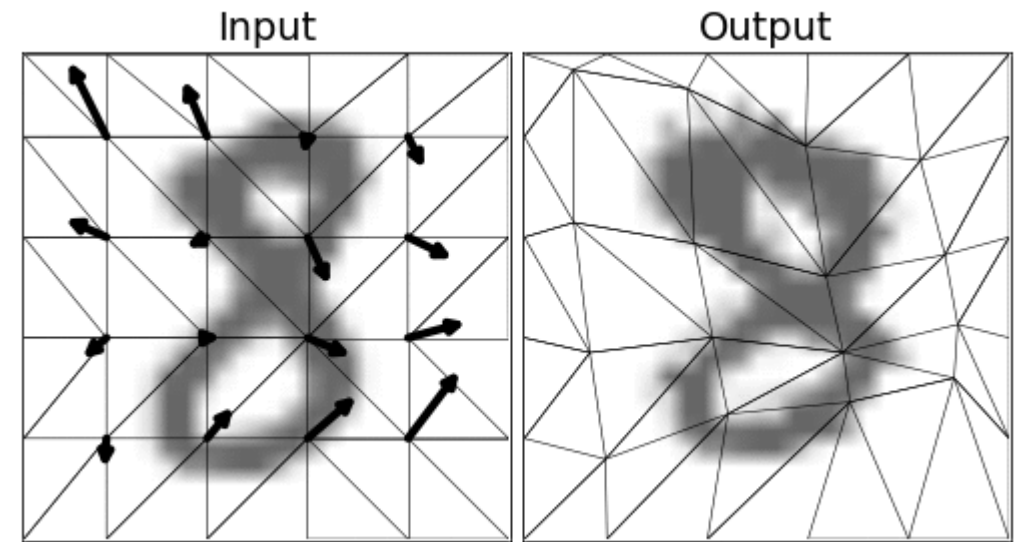


그림. 2 숫자 8이 표현된 원본 그림(좌)과 elastic distortion이 적용된 그림(우)

- 기존 연구들은 서로 다른 Transformation들을 결합하기 위한 최적의 전략을 찾는 것에 집중
- AutoAugment는 RNN 모델을 통해 데이터로부터 데이터 증강 규칙을 학습함
  - Transformation의 순서 뿐만 아니라 적용 probability와 magnitude를 선택
- AutoAugment에서 규칙의 적용은 여러 단계의 확률 계층으로 구성함
  - 모든 미니배치의 모든 이미지에 대해서 sub-policy는 균일 확률로 선택됨
  - 각 sub-policy안의 Transformation들은 각각의 적용 확률과 강도를 가짐
  - 일부 Transformation들은 방향에 대한 확률을 가짐
- Transformation은 적용될 probability와 magnitude에 대한 parameter를 가짐
  - 0.1 = probability(확률)
  - 7 = magnitude (강도)

```
10 self.policies = [  
11     ['Invert', 0.1, 7, 'Contrast', 0.2, 6],  
12     ['Rotate', 0.7, 2, 'TranslateX', 0.3, 9],  
13     ['Sharpness', 0.8, 1, 'Sharpness', 0.9, 3],  
14     ['ShearY', 0.5, 8, 'TranslateY', 0.7, 9],  
15     ['AutoContrast', 0.5, 8, 'Equalize', 0.9, 2],  
16     ['ShearY', 0.2, 7, 'Posterize', 0.3, 7],  
17     ['Color', 0.4, 3, 'Brightness', 0.6, 7],  
18     ['Sharpness', 0.3, 9, 'Brightness', 0.7, 9],  
19     ['Equalize', 0.6, 5, 'Equalize', 0.5, 1],  
20     ['Contrast', 0.6, 7, 'Sharpness', 0.6, 5],  
21     ['Color', 0.7, 7, 'TranslateX', 0.5, 8],  
22     ['Equalize', 0.3, 7, 'AutoContrast', 0.4, 8],  
23     ['TranslateY', 0.4, 3, 'Sharpness', 0.2, 6],
```

그림. 1 데이터 증강에 사용되는 sub-policy와 operation 그리고 각 operation들의 확률과 강도

- 확률의 계층은 통해 모델이 훈련되는 **표현의 다양성을 증가**
  - 데이터셋을 통한 일반화 성능을 크게 향상시킴
- PBA와 Fast AutoAugment는 기존 방법의 검색 공간과 더 효율적인 규칙을 찾기 위해 향상된 최적화 알고리즘을 제안
  - 규칙 탐색에 걸리는 **시간이 크게 단축**되었지만, 탐색 과정이 **별도의 검색 단계에서 구현**
    - 별도의 검색 과정이 갖는 단점이 남아있음
      - 추가적인 연산 필요, 다른 애플리케이션과 도메인에 적용하기 어려움



- RandAugment는 분리된 탐색 과정의 필요성을 제거하는 것을 목표
  - 분리된 탐색 과정은 학습을 매우 복잡하게 만들고 계산 복잡도를 증가
  - 특히, proxy task에서 얻은 결과는 sub-optimal의 결과
- 분리된 탐색 과정을 제거하기 위해서, 저자들은 데이터 증강에 사용되는 parameter들을 모델 학습을 위한 parameter로 전환
  - 이전의 학습을 통한 데이터 증강 방법들은 30개 이상의 parameter를 가지고 있으며, 저자들은 데이터 증강에 사용되는 parameter의 수를 줄이는 것에 집중
  - Transformation에서 사용되는 probability와 magnitude에 사용되는 parameter를 줄임

- 이전의 연구들은 학습을 통한 데이터 증강 방법의 주요 이점은 데이터 샘플의 다양성의 증가로부터 발생한다는 것을 발견
  - 실제로, 이전 연구들은 사용 가능한 transformation과 각 transformation이 적용될 **probability**와 **magnitude**를 선택하는 측면에서 **수많은 규칙을 열거**
- 본 저자들은 데이터의 다양성을 유지한 상태에서 **parameter space**를 줄임

- 학습된 규칙과 각 transformation이 적용될 확률을 **균일 확률  $\frac{1}{K}$** 로 대체함으로써 **parameter가 필요하지 않도록 처리**

- 하나의 데이터에 대한 N개의 transformation이 주어졌을 때, RandAugment는  $K^N$ 개의 잠재적인 규칙을 표현

• identity	• autoContrast	• equalize
• rotate	• solarize	• color
• posterize	• contrast	• brightness
• sharpness	• shear-x	• shear-y
• translate-x	• translate-y	

```
10 self.policies = [  
11     ['Invert', 0.1, 7, 'Contrast', 0.2, 6],  
12     ['Rotate', 0.7, 2, 'TranslateX', 0.3, 9],  
13     ['Sharpness', 0.8, 1, 'Sharpness', 0.9, 3],  
14     ['ShearY', 0.5, 8, 'TranslateY', 0.7, 9],  
15     ['AutoContrast', 0.5, 8, 'Equalize', 0.9, 2],  
16     ['ShearY', 0.2, 7, 'Posterize', 0.3, 7],  
17     ['Color', 0.4, 3, 'Brightness', 0.6, 7],  
18     ['Sharpness', 0.3, 9, 'Brightness', 0.7, 9],  
19     ['Equalize', 0.6, 5, 'Equalize', 0.5, 1],  
20     ['Contrast', 0.6, 7, 'Sharpness', 0.6, 5],  
21     ['Color', 0.7, 7, 'TranslateX', 0.5, 8],  
22     ['Equalize', 0.3, 7, 'AutoContrast', 0.4, 8],  
23     ['TranslateY', 0.4, 3, 'Sharpness', 0.2, 6],
```

그림. 1 데이터 증강에 사용되는 sub-policy와 operation 그리고 각 operation들의 확률과 강도

# RandAugment: magnitude

10/21

- Transformation의 적용 강도를 나타내는 magnitude parameter에서 사용되는 parameter를 최소화
- Transformation의 magnitude를 표현하기 위해서 AutoAugment와 동일한 선형 크기를 사용
  - Magnitude는 0부터 10까지의 정수 크기를 사용하고 10은 변환 강도의 최대 강도를 의미
  - 데이터 증강 규칙은 각 증강에 대한 정수를 식별하는 것으로 구성됨
- 각 transformation에 대한 학습된 transformation magnitude parameter가 모델의 학습 동안에 유사한 **schedule**을 따른다는 것을 발견하였고 **single global distortion M**을 사용해서 모든 transformation을 parameterizing 하기에 충분하다고 가정함

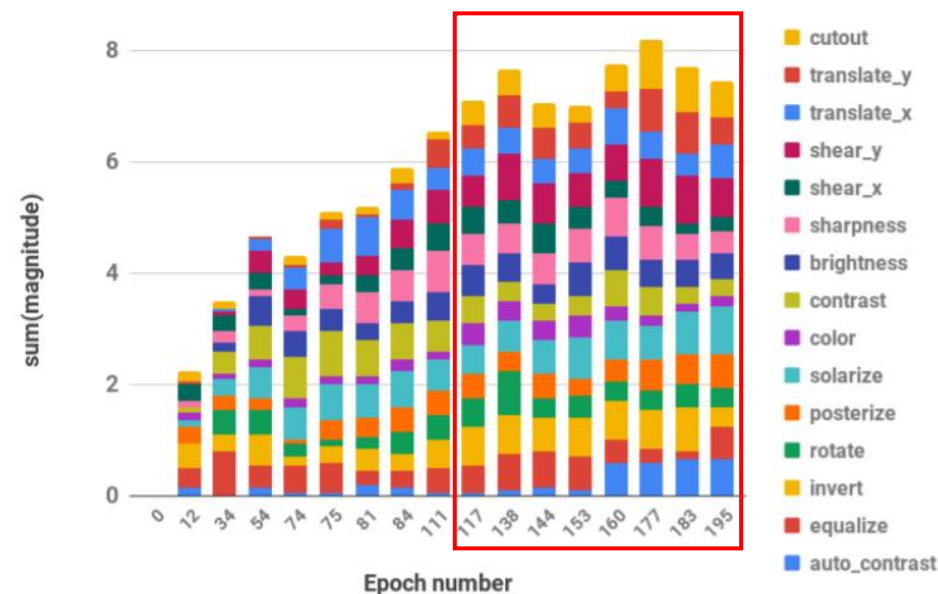


그림. 1 데이터 증강 규칙을 학습하는 과정에서의 magnitude parameter의 변화

# RandAugment: magnitude (계속)

11/21

- **M**을 선택하기 위해서 4가지 방법에 대해서 실험을 진행
  - A random magnitude, a **constant magnitude**, a linearly increasing magnitude and a random magnitude with increasing upper bound.
- 위 실험에서 모든 방법들에 대한 결과가 잘 나오는 것을 확인
  - 하나의 hyperparameter를 갖는 **constant magnitude**를 선택

Magnitude Method	Accuracy
Random Magnitude	97.3
Constant Magnitude	97.2
Linearly Increasing Magnitude	97.2
Random Magnitude with Increasing Upper Bound	97.3

Table 7. **Results for different ways of setting the global magnitude parameter  $M$ .** All magnitude methods were run on CIFAR-10 with Wide-ResNet-28-10 for 200 epochs. The reported accuracy is the average of 10 runs on the validation set for the best hyperparameter setting for that magnitude method. All magnitude methods searched over had 48 different hyperparameter settings tried.

# RandAugment: code results

12/21

- 제안하는 방법은 N과 M 두 개의 parameter를 사용하고, Python에서 두 줄로 구현될 정도로 단순
  - 두 parameter N과 M은 크기가 커질수록 강도가 강해진다는 것을 알 수 있듯이 human-interpretable함
- Search space가 매우 작기 때문에 단순한 **grid search** 만으로도 매우 효과적이라는 것을 발견

```
transforms = [  
    'Identity', 'AutoContrast', 'Equalize',  
    'Rotate', 'Solarize', 'Color', 'Posterize',  
    'Contrast', 'Brightness', 'Sharpness',  
    'ShearX', 'ShearY', 'TranslateX', 'TranslateY']  
  
def randaugment (N, M):  
    """Generate a set of distortions.  
  
    Args:  
        N: Number of augmentation transformations to  
            apply sequentially.  
        M: Magnitude for all the transformations.  
    """  
  
    sampled_ops = np.random.choice(transforms, N)  
    return [(op, M) for op in sampled_ops]
```

그림. 1 Numpy를 기반으로 한 RandAugment에 대한 Python code

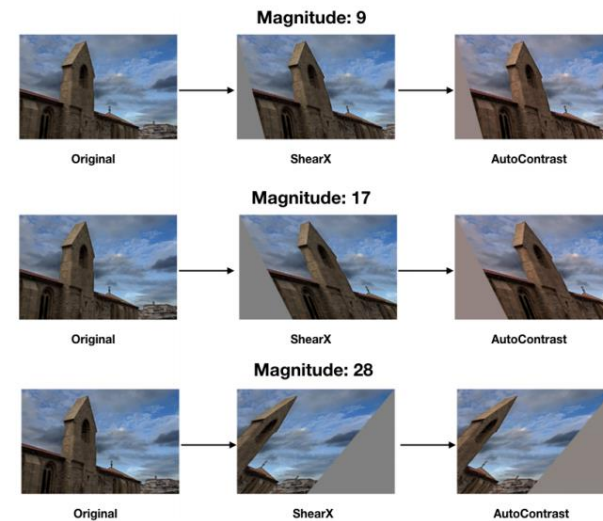


그림. 2 RandAugment를 통해 증강한 데이터의 예시

# RandAugment: separate proxy task

13/21

- 학습을 통한 데이터 증강의 중심 전제는 더 큰 작업을 반영 할 수 있는 작은 프록시 작업을 구성하는 것임
  - 학습 된 증가 정책은 모델의 성능을 향상시키기는 하지만 large target task에 대해 최적의 데이터 증가 규칙을 학습하기는 어려움
- Small proxy task에서 학습한 최적의 증강 규칙이 large target task에서도 최적의 결과를 보여준다는 가설에 대해서 실험을 통해 반박
  - 특히, small proxy task의 성능을 향상시키기 위해 일반적으로 제한되는 두 가지 요소인 model size와 dataset size를 중심으로 실험을 진행

# Systematic failures of a separate proxy task

14/21

- 좌측 두 사진은 model size에 따른 model accuracy와 distortion magnitude의 변화를 표현 (빨간색 > 보라색 > 파란색)
  - Model size가 커질수록 최적의 distortion magnitude의 크기도 커지는 것을 알 수 있음
  - 네트워크 크기의 증가함에 따라 최적의 변환 강도가 단조로운 추세를 보임
- 우측 두 사진은 dataset size에 따른 model accuracy와 distortion magnitude의 변화를 표현 (빨간색 > 보라색 > 파란색)
  - Dataset의 크기와 상관없이 데이터 증강을 통해 성능을 개선할 수 있는 것을 관찰함
  - 특히, 대규모 dataset으로 학습된 model의 최적 distortion magnitude가 더 크다는 것을 발견
    - 직관에 반하는 결과는 소규모 데이터셋에 데이터 증강을 공격적으로 할 경우 signal-to-noise ratio가 낮아서 발생하는 문제라고 추측하고 있음
- 본 실험의 결과로 proxy task에서 학습한 transformation policy는 large target task에 최적의 결과가 아님

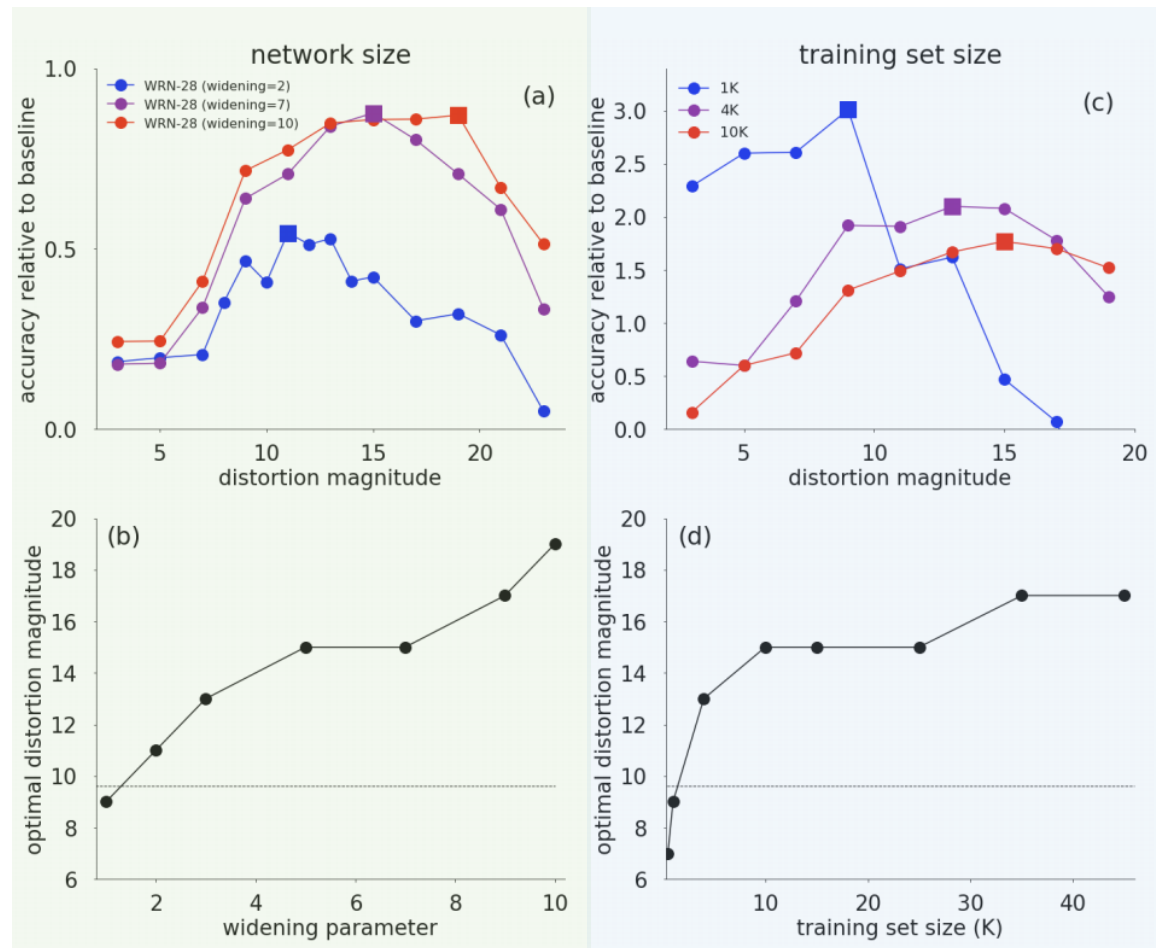


그림. 1 Numpy를 기반으로 한 RandAugment에 대한 Python code



# CIFAR&SVHN results

15/21

- CIFAR
  - 기존의 방법들과 비교해서 경쟁력 있거나 더 우수한 성능을 보임
- SVHN
  - SVHN은 자연 영상이 아닌 숫자로 구성된 번호판 영상이기 때문에, 자연 영상인 CIFAR 데이터와 비교했을 때 데이터 증강 전략이 다름
    - 실제로, AutoAugment는 CIFAR와 SVHN에 학습에 사용된 데이터 증강 규칙이 다른 것을 발견
- SVHN와 CIFAR의 데이터 특성의 차이에도 불구하고 RandAugment는 이전에 사용된 transformation의 변화 없이도 다른 방법들과 유사하거나 더 우수한 성능을 보임

	baseline	PBA	Fast AA	AA	RA
<b>CIFAR-10</b>					
Wide-ResNet-28-2	94.9	-	-	<b>95.9</b>	95.8
Wide-ResNet-28-10	96.1	<b>97.4</b>	97.3	<b>97.4</b>	97.3
Shake-Shake	97.1	<b>98.0</b>	<b>98.0</b>	<b>98.0</b>	<b>98.0</b>
PyramidNet	97.3	<b>98.5</b>	98.3	<b>98.5</b>	<b>98.5</b>
<b>CIFAR-100</b>					
Wide-ResNet-28-2	75.4	-	-	<b>78.5</b>	78.3
Wide-ResNet-28-10	81.2	<b>83.3</b>	82.7	82.9	<b>83.3</b>
<b>SVHN (core set)</b>					
Wide-ResNet-28-2	96.7	-	-	98.0	<b>98.3</b>
Wide-ResNet-28-10	96.9	-	-	98.1	<b>98.3</b>
<b>SVHN</b>					
Wide-ResNet-28-2	98.2	-	-	<b>98.7</b>	<b>98.7</b>
Wide-ResNet-28-10	98.5	98.9	98.8	98.9	<b>99.0</b>

비행기  
자동차  
새  
고양이  
사슴  
개  
개구리  
말  
배  
트럭

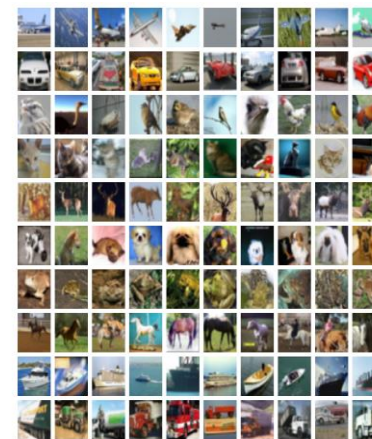


그림. 1 CIFAR10의 데이터 구성



그림. 2 SVHN의 데이터 구성



- CIFAR와 SVHN를 학습한 모델에 사용된 데이터 증강 방법은 ImageNet과 같은 large-scale task에서 동일하게 성능을 향상시켜주지 않음
- AutoAugment는 다른 작업과 다르게 ImageNet에서 성능 향상이 없으며, 특히 거대한 모델에서 두드러짐
  - 데이터 증강 규칙을 학습하기 위해서 1000개의 ImageNet class 중에서 10%로 small proxy task의 작업을 제한하였기 때문이라고 추측

	baseline	Fast AA	AA	RA
ResNet-50	76.3 / 93.1	<b>77.6 / 93.7</b>	<b>77.6 / 93.8</b>	<b>77.6 / 93.8</b>
EfficientNet-B5	83.2 / 96.7	-	83.3 / 96.7	<b>83.9 / 96.8</b>
EfficientNet-B7	84.0 / 96.9	-	84.4 / 97.1	<b>85.0 / 97.2</b>

Table 3. **ImageNet results.** Top-1 and Top-5 accuracies (%) on ImageNet. Baseline and AutoAugment (AA) results on ResNet-50 are from [5]. Fast AutoAugment (Fast AA) results are from [25]. EfficientNet results with and without AutoAugment are from [47]. Highest accuracy for each model is presented in bold. Note that Population Based Augmentation (PBA) [20] has not been implemented on ImageNet.

- AutoAugment와 유사한 성능을 보임
  - AutoAugment는 RandAugment에 없는 **bounding box에 적절한 transformation을 추가적으로 사용**
  - AutoAugment는 증강 규칙을 찾기 위해 GPU를 사용하여 15K 시간동안 탐색한 반면 RandAugment는 **6개의 hyperparameter를 사용하여 조정**
- RandAugment의 성능을 잠재적으로 개선할 수 있도록 bounding box에 특정한 transformation을 추가하는 것을 추후 연구로 남김

model	augmentation	mAP	search space
ResNet-101	Baseline	38.8	0
	AutoAugment	<b>40.4</b>	$10^{34}$
	RandAugment	40.1	$10^2$
ResNet-200	Baseline	39.9	0
	AutoAugment	<b>42.1</b>	$10^{34}$
	RandAugment	41.9	$10^2$

Table 4. **Results on object detection.** Mean average precision (mAP) on COCO detection task. Higher is better. Search space size is reported as the order of magnitude of the number of possible augmentation policies. Models are trained for 300 epochs from random initialization following [57].

- 각각의 Transformation에 대한 개별적인 효과를 알아보기 위해 랜덤하게 구성된 transformation list에 임의의 transformation을 추가하고 이것에 대한 검증 정확도의 평균을 계산
  - Rotate, shear와 같은 기하학적인 transformation이 적용되었을 때 성능향상을 보임
  - RandAugment가 transformation의 선택에는 대체로 무감각함

transformation	$\Delta$ (%)	transformation	$\Delta$ (%)
rotate	<b>1.3</b>	shear-x	0.9
shear-y	0.9	translate-y	0.4
translate-x	0.4	autoContrast	0.1
sharpness	0.1	identity	0.1
contrast	0.0	color	0.0
brightness	0.0	equalize	-0.0
solarize	-0.1	posterize	-0.3

Table 5. **Average improvement due to each transformation.** Average difference in validation accuracy (%) when a particular transformation is added to a randomly sampled set of transformations. For this ablation study, Wide-ResNet-28-2 models were trained on CIFAR-10 using RandAugment ( $N = 3$ ,  $M = 4$ ) with the randomly sampled set of transformations, with no other data augmentation.

- 데이터 증강은 데이터 기반의 머신러닝 알고리즘에서 state-of-the-art 성능을 달성하기 위해서 필수적으로 사용되는 방법
  - 학습을 통한 데이터 증강 방법은 설계 전략을 자동화하는 것을 도와주었으며, state-of-the-art 성능을 보임
- RandAugment는 CIFAR, SVHN, ImageNet 그리고 COCO에서 증강 규칙을 학습하기 위한 **분리된 검색 과정 없이 기존**  
**의 데이터 증강 알고리즘과 유사하거나 뛰어넘는 성능을 보임**
- 이전의 연구에서는 학습된 데이터 증강 규칙을 더 큰 데이터나 모델에 확장시키는 것은 어려운 문제
  - 기존의 연구들은 오직 작은 모델과 데이터의 일부분을 통해서 증강 규칙을 최적화 할 수 있음
- 제안하는 방법은 적은 계산 비용으로 모델의 성능을 향상시켰으며, ImageNet과 COCO와 같이 큰 데이터에서도 적용이 가능
  - RandAugment는 동일한 증강 규칙을 사용하여 서로 다른 task 또는 domain에서 좋은 결과를 보임

QnA

- Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501, 2018.
- Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. arXiv preprint arXiv:1905.00397, 2019.
- Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. Population based augmentation: Efficient learning of augmentation policy schedules. arXiv preprint arXiv:1905.05393, 2019.
- Barret Zoph, Ekin D Cubuk, Golnaz Ghiasi, Tsung-Yi Lin, Jonathon Shlens, and Quoc V Le. Learning data augmentation strategies for object detection. arXiv preprint arXiv:1906.11172, 2019.

감사합니다.