Circumventing Outliers of AutoAugment with Knowledge Distillation

Longhui Wei[1,2⋆], An Xiao[1⋆], Lingxi Xie[1], Xiaopeng Zhang[1], Xin Chen[1,3], Qi Tian[1]

[1]Huawei Inc., [2]University of Science and Technology of China, [3]Tongji University
weilh2568@gmail.com, xiaoan1@huawei.com, 198808xc@gmail.com
1410452@tongji.edu.cn, zxphistory@gmail.com, tian.qi1@huawei.com

Abstract. AutoAugment has been a powerful algorithm that improves the accuracy of many vision tasks, yet it is sensitive to the operator space as well as hyper-parameters, and an improper setting may degenerate network optimization. This paper delves deep into the working mechanism, and reveals that AutoAugment may remove part of discriminative information from the training image and so insisting on the ground-truth label is no longer the best option. To relieve the inaccuracy of supervision, we make use of knowledge distillation that refers to the output of a teacher model to guide network training. Experiments are performed in standard image classification benchmarks, and demonstrate the effectiveness of our approach in suppressing noise of data augmentation and stabilizing training. Upon the cooperation of knowledge distillation and AutoAugment, we claim the **new state-of-the-art** on ImageNet classification with a top-1 accuracy of **85.8**%.

Keywords: AutoML, AutoAugment, Knowledge Distillation

1 Introduction

Automated machine learning (AutoML) has been attracting increasing attentions in recent years. In standard image classification tasks, there are mainly two categories of AutoML techniques, namely, neural architecture search (NAS) and hyper-parameter optimization (HPO), both of which focus on the possibility of using automatically learned strategies to replace human expertise. AutoAugment [4] belongs to the latter, which goes one step beyond conventional data augmentation techniques (*e.g.*, horizontal flipping, image rescaling & cropping, color jittering, *etc.*) and tries to combine them towards generating more training data without labeling new images. It has achieved consistent accuracy gain in image classification [4], object detection [11], *etc.*, and meanwhile efficient variants of AutoAugment have been proposed to reduce the computational burden in the search stage [17, 25, 13].

Despite their ability in improving recognition accuracy, we note that AutoAugment-based methods often require the search space to be well-designed. Without

⋆ The first two authors contributed equally to this work.

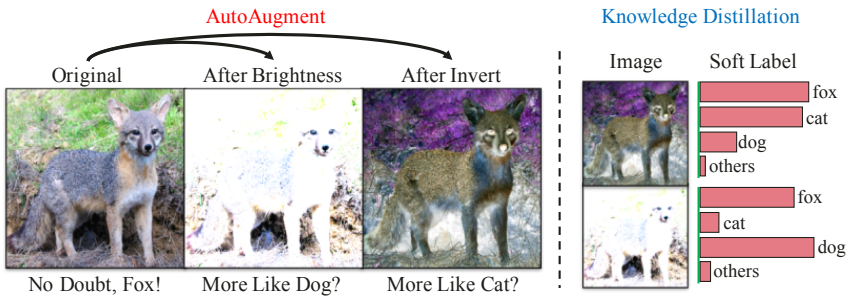


Fig. 1. Left: an image and its augmented copies generated by AutoAugment. The original image is clean and there is no doubt to use the ground-truth label, while the augmented counterparts look more like other classes which the annotation is not aware of. This phenomenon is called **augment ambiguity**. **Right**: We leverage the idea of knowledge distillation to provide softened signals to avoid ambiguity.

careful control (*e.g.*, in an expanded search space or with an increased distortion magnitude), these methods are not guaranteed to perform well – as we shall see in Section 3.2, an improper hyper-parameter may deteriorate the optimization process, resulting in even lower accuracy compared to the baseline. This puts forward a hard choice between more information (seeing a wider range of augmented images) and safer supervision (restricting the augmented image within a relatively small neighborhood around the clean image), which downgrades the upper-bound of AutoAugment-based methods.

In this paper, we investigate the reason of this contradictory. We find that when heavy data augmentation is added to the training image, it is probable that part of its semantic information is removed. An example of changing image brightness is shown in Figure 1, and other transformation such as image translation and shearing can also incur information loss and make the image class unrecognizable (refer to Figure 3). We name this phenomenon **augment ambiguity**. In such contaminated training images, insisting on the ground-truth label is no longer the best choice, as the inconsistency between input and supervision can confuse the network. Intuitively, complementary information that relates the augmented image to similar classes may serve as a better source of supervision.

Motivated by the above, we leverage the idea of knowledge distillation which uses a standalone model (often referred to as the *teacher*) to guide the target network (often referred to as the *student*). For each augmented image, the student receives supervision from both the ground-truth and the teacher signal, and in case that part of semantic information is removed, *e.g.*, an ambiguous (Figure 1) or an out-of-distribution (Figure 3) instance, the teacher can provide softened labels to avoid confusion. The extra loss between the teacher and student is measured by the KL-divergence between the score distributions of their top-ranked classes, and the number of involved classes is positively correlated to the magnitude of augmentation, since a larger magnitude often eliminates more semantics and causes smoother score distributions.

The main contribution of this paper is to reveal that *knowledge distillation is a natural complement to uncontrolled data augmentation, such as AutoAugment and its variants*. The effectiveness of our approach is verified in the space of AutoAugment [4] as well as that of RandAugment [5] with different strengths of transformations. Knowledge distillation brings consistent accuracy gain to recognition, in particular when the distortion magnitude becomes larger. Experiments are performed on standard image classification benchmarks, namely, CIFAR-10/100 and ImageNet. On CIFAR-100, with a strong baseline of PyramidNet [12] and ShakeDrop [49] regularization, we achieve a test error of 10.6%, outperforming all competitors with similar training costs. On ImageNet, in the RandAugment space, we boost the top-1 accuracy of EfficientNet-B7 [41] from 84.9% to 85.5%, with a significant improvement of 0.6%. Note that without knowledge distillation, RandAugment with a large distortion magnitude may suffer unstable training. Moreover, on top of EfficientNet-B8 [45], we set a **new record** on ImageNet classification (without extra training data) by claiming a top-1 accuracy of **85.8**%, surpassing the previous best by a non-trivial margin.

2 Related Work

Deep learning [24], in particular training deep neural networks, has been the standard methodology in computer vision. Modern neural networks, either manually-designed [22, 34, 38, 14, 19] or automatically searched [58, 31, 59, 27, 40, 41], often contain a very large number of trainable parameters and thus raise the challenge of collecting more labeled data to avoid over-fitting. Data augmentation is a standard strategy to generate training data without additional labeling costs. Popular options of transformation include horizontal flipping, color/contrast jittering, image rotation/shearing, *etc.*, each of which slightly alters the geometry and/or pattern of an image but keeps its semantics (*e.g.*, the class label) unchanged. Data augmentation has been verified successful in a wide range of visual recognition tasks, including image classification [7, 54, 52], object detection [35], semantic segmentation [8], person re-identification [57], *etc.* Researchers have also discussed the connection between data augmentation and network regularization [36, 10, 49] methods.

With the rapid development of automated machine learning (AutoML) [43], researchers proposed to learn data augmentation strategies in a large search space to replace the conventional hand-designed augmentation policies. AutoAugment [4] is one of the early efforts that works on this direction. It first designed a search space with a number of transformations and then applied reinforcement learning to search for powerful combinations of the transformations to arrive at a high validation accuracy. To alleviate the heavy computational costs in the search stage, researchers designed a few efficient variants of AutoAugment. Fast AutoAugment [25] moved the costly search stage from training to evaluation through bayesian optimization, and population-based augmentation [17] applied evolutionary algorithms to generate policy schedule by only a single run of 16 child models. Online hyper-parameter learning [26] combined the

search stage and the network training process, and faster AutoAugment [13] formulated the search process into a differentiable function, following the recently emerging weight-sharing NAS approaches [2, 30, 28]. Meanwhile, some properties of AutoAugment have been investigated, such as whether aggressive transformations need to be considered [17] and how transformations of enriched knowledge are effectively chosen [55]. Recently, RandAugment [5] shared another opinion that the search space itself may have contributed most: based on a well-designed set of transformations, a random policy of augmentation works sufficiently well.

Knowledge distillation was first introduced as an approach to assist network optimization [16]. The goal is to improve the performance of a target network (often referred to as the student model) using two sources of supervision, one from the ground-truth label, and the other from the output signal of a pre-trained network (often referred to as the teacher model). Beyond its wide application on model compression (large teacher, small student [33, 16]) and model initialization (small teacher, large student [34, 3]), researchers later proposed to use it for standard network training, with the teacher and student models sharing the same network architecture [9, 50, 51], and sometimes under the setting of semi-supervised learning [42, 56]. There have been discussions on the working mechanism of knowledge distillation, and researchers advocated for the so-called 'dark knowledge' [16] being some kind of auxiliary supervision, obtained from the pre-trained model [50, 1] and thus different from the softened signals based on label smoothing [39, 29].

In this paper, we build the connection between knowledge distillation and AutoAugment by showing that the former is a natural complement to the latter, which filters out noises introduced by overly aggressive transformations.

3 Our Approach

3.1 Preliminaries: Data Augmentation with AutoML

Let $\mathcal{D} = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ be a labeled image classification dataset with N samples, in which $\mathbf{x}_n$ denotes the raw pixels and y_n denotes the annotated label within the range of $\{1, 2, \ldots, C\}$, C is the number of classes. $\mathbf{y}_n$ is the vectorized version of y_n with the dimension corresponding to the true class assigned a value of 1 and all others being 0. The goal is to train a deep network, $\mathbb{M} : \mathbf{y}_n = \mathbf{f}(\mathbf{x}_n; \boldsymbol{\theta})$, where $\boldsymbol{\theta}$ denotes the trainable parameters. The dimensionality of $\boldsymbol{\theta}$ is often very large, *e.g.*, tens of millions, exceeding the size of dataset, N, in most of cases. Therefore, network training is often a ill-posed optimization problem and incurs over-fitting without well-designed training strategies.

The goal of data augmentation is to enlarge the set of training images without actually collecting and annotating more data. It starts with defining a transformation function, $\mathbf{x}_n^{\boldsymbol{\tau}} \doteq \mathbf{g}(\mathbf{x}_n, \boldsymbol{\tau})$, in which $\boldsymbol{\tau} \sim \mathcal{T}$ is a multi-dimensional vector parameterizing how the transformations are performed. Note that each dimension of $\boldsymbol{\tau}$ can take either discrete (*e.g.*, whether the image is horizontally flipped) or continuous (*e.g.*, the image is rotated for a specific angle), and different transformations can be applied to an image towards richer combinations. The idea of

AutoAugment [4] is to optimize the distribution, $\mathcal{T}$, so that the model optimized on the augmented training set performs well on the validation set:

$$\mathcal{T}^\star = \arg\min_{\mathcal{T}} \mathcal{L}(\mathbf{g}(\mathbf{x}_n; \boldsymbol{\tau} \sim \mathcal{T}), \mathbf{y}_n; \boldsymbol{\theta}^\star_{\mathcal{T}} \mid (\mathbf{x}_n, \mathbf{y}_n) \sim \mathcal{D}_{\text{val}}), \quad (1)$$

$$\text{in which} \quad \boldsymbol{\theta}^\star_{\mathcal{T}} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}(\mathbf{g}(\mathbf{x}_n; \boldsymbol{\tau} \sim \mathcal{T}), \mathbf{y}_n; \boldsymbol{\theta} \mid (\mathbf{x}_n, \mathbf{y}_n) \sim \mathcal{D}_{\text{train}}).$$

Here, $\mathcal{D}_{\text{train}}$ and $\mathcal{D}_{\text{val}}$ are two subsets of $\mathcal{D}$, used for training and validating the quality of $\mathcal{T}$, respectively. The loss function follows any conventions, *e.g.*, the cross-entropy form, $\mathcal{L}(\mathbf{x}^{\boldsymbol{\tau}}_n, \mathbf{y}_n; \boldsymbol{\theta}) = \mathbf{y}_n^\top \cdot \ln \mathbf{f}(\mathbf{x}^{\boldsymbol{\tau}}_n; \boldsymbol{\theta})$. Eqn (1) is a two-stage optimization problem, for which existing approaches either applied reinforcement learning [4, 25, 17] or weight-sharing methods [13] which are often more efficient.

We follow the convention to assign each dimension in $\boldsymbol{\tau}$ to be an individual transformation, with the complete list shown below:

- `invert`
- `autoContrast`
- `equalize`
- `rotate`
- `solarize`
- `color`
- `posterize`
- `contrast`
- `brightness`
- `sharpness`
- `shear-x`
- `shear-y`
- `translate-x`
- `translate-y`

Therefore, $\boldsymbol{\tau}$ is a 14-dimensional vector and each dimension of $\boldsymbol{\tau}$ represents the magnitude of the corresponding transformation. For example, the fourth dimension of $\boldsymbol{\tau}$ represents the magnitude of `rotate` transformation, and a value of zero indicates the corresponding transformation being switched off. Each time a transformation is sampled from the distribution, $\boldsymbol{\tau} \sim \mathcal{T}$, at most two dimensions in it are set to be non-zero, and each selected transformation is assigned a probability that it is applied after each training image is sampled online.

3.2 AutoAugment Introduces Noisy Training Images

AutoAugment makes it possible to generate infinitely many images which do not exist in the original training set. On the upside, this reduces the risk of over-fitting during the training process; on the downside, it can introduce a considerable amount of outliers to the training process. Typical examples are shown in Figure 2. When an image with its upper part occupied by main content (*e.g.*, *bee*) is sampled, the transformation of `translate-y` (shifting the image along the vertical direction) suffers risk of removing all discriminative contents within it outside the visible area, and thus the augmented image becomes meaningless in semantics. Nonetheless, the training process is not always aware of such noises and still uses the ground-truth signal, a one-hot vector, to supervise and thus confuse the deep network.

In Figure 2, we also show how the training loss and validation accuracy curves change along with the magnitude of transformation. When the magnitude is 0 (*i.e.*, no augmentation is used), it is easy for the network to fit the training set and thus the training loss quickly drops, but the validation accuracy remains low which indicates over-fitting. With a relatively low magnitude of augmentation, the training loss increases gradually meanwhile the validation accuracy arrives at a higher plateau, *i.e.*, over-fitting is alleviated. However, if the magnitude of augmentation continues growing, it becomes more and more difficult

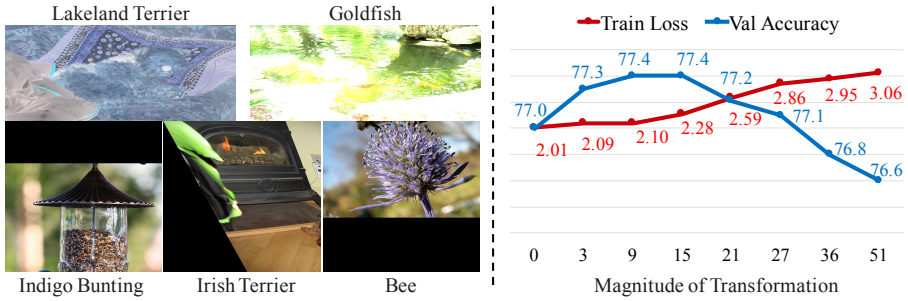


Fig. 2. Left: AutoAugment can generate meaningless training images but still assigns deterministic class labels to them. **Right**: The results of EfficientNet-B0 with different magnitudes of transformation on ImageNet. The training difficulty increases gradually with enlarging the magnitude of transformation, while the validation accuracy rises initially but drops at last. This phenomenon reveals the model starts from over-fitting to under-fitting.

to fit the training set, *i.e.*, the model suffers under-fitting. In particular, when the magnitude is set to be 36, the noisy data introduced to the training set is sufficiently high to bias the model training, *i.e.*, the results is lower than the baseline without AutoAugment.

From the above analysis, we realize that AutoAugment is indeed balancing between richer training data and heavier noises. Researchers provided comments from two aspects: some of them argued that the transformation strategies may have been overly aggressive and need to be controlled [17], while some others advocated for the benefit of exploring aggressive transformations so that richer information is integrated into the trained model [55]. We deal with this issue from a new perspective. We believe that aggressive transformations are useful to training, yet treating all augmented images just like they are clean (non-augmented) samples is not the optimal choice. Moreover, the same transformations operated on different images will cause different results, *i.e.*, some generated images can enrich the diversity of training set but the others are biased. Therefore, we treat every image differently for preserving richer information but filtering out noises.

3.3 Circumventing Outliers with Knowledge Distillation

Our idea is very simple. For a training image generated by AutoAugment, $\mathbf{g}(\mathbf{x}_n;\boldsymbol{\tau})$, we provide two-source supervision signals to guide network optimization. The first one remains the same as the original training process, with the standard cross-entropy loss computed based on the ground-truth class, $\mathbf{y}_n$. The second one comes from a pre-trained model which provides an individual judgment of $\mathbf{g}(\mathbf{x}_n;\boldsymbol{\tau})$, *i.e.*, whether it contains sufficient semantics for classification. Let $\mathbb{M}^{\mathrm{T}}$ and $\mathbb{M}^{\mathrm{S}}$ denote the pre-trained (teacher) and target (student) model, where the superscripts of T and S represent 'teacher' and 'student', respectively,

and thus the network loss function is upgraded to be:

$$\mathcal{L}^{\mathrm{KD}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}, \mathbf{y}_n; \boldsymbol{\theta}^{\mathrm{S}}\right) = \mathbf{y}_n^{\top} \cdot \ln \mathbf{f}^{\mathrm{S}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{S}}\right) + \lambda \cdot \mathrm{KL}\Big[\mathbf{f}^{\mathrm{S}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{S}}\right) \| \mathbf{f}^{\mathrm{T}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{T}}\right)\Big], \tag{2}$$

where λ is the balancing coefficient, and we have followed the convention to use the KL-divergence to compute the distance between teacher and student outputs, two probabilistic distributions over all classes.

Intuitively, when the semantic information of an image is damaged by data augmentation, the teacher model that is 'unaware' of augmentation should produce less confident probabilistic outputs, *e.g.*, if an original image, $\mathbf{x}_n$, contains a specific kind of *bird* and some parts of the *bird* is missing or contaminated by augmentation, $\boldsymbol{\tau}$, then we expect the probabilistic scores of the augmented image, $\mathbf{x}_n^{\boldsymbol{\tau}}$, to be distributed over a few classes with close relationship to the true one. We introduce a hyper-parameter, K, and consider the K classes with the highest scores in $\mathbf{f}^{\mathrm{T}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{T}}\right)$, forming a set denoted by $\mathcal{C}_K\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{T}}\right)$. Due to the reason that the class probability decays rapidly with ranking, and low-ranked scores may contain much noise, it is often unsafe to force the student model to fit all teacher scores, so most often, we have $K \ll C$, and the choice of K will be discussed empirically in the experimental section. The KL-divergence between $\mathbf{f}^{\mathrm{T}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{T}}\right)$ and $\mathbf{f}^{\mathrm{S}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{S}}\right)$ is thus modified as:

$$\mathrm{KL}\Big[\mathbf{f}^{\mathrm{S}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{S}}\right) \| \mathbf{f}^{\mathrm{T}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{T}}\right)\Big] = \sum_{c \in \mathcal{C}_K\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{T}}\right)} f_c^{\mathrm{T}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{T}}\right) \cdot \ln \frac{f_c^{\mathrm{S}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{S}}\right)}{f_c^{\mathrm{T}}\left(\mathbf{x}_n^{\boldsymbol{\tau}}; \boldsymbol{\theta}^{\mathrm{T}}\right)}, \tag{3}$$

where f_c denotes the c-th dimension of $\mathbf{f}$.

3.4 Discussions and Relationship to Prior Work

A few prior work [1, 50] studied how knowledge distillation works in the scenarios that teacher and student models have the same capacity. They argued that the teacher model should be strong enough so as not to provide low-quality supervision to the student model. However, this work provides a novel usage of the teacher signal: suppressing noises introduced by data augmentation. From this perspective, the teacher model can be considerably weaker than the student model but still contribute to recognition accuracy. Experimental results on CIFAR-100 (setting and details are provided in Section 4.1) show that a pre-trained Wide-ResNet-28-10 [53] with AutoAugment (test set error rate of 17.1%) can reduce the test set error rate of a Shake-Shake (26 2x96D) [10] trained with AutoAugment from 14.3% to 13.8%.

We noticed prior work [15] argued that data augmentation may introduce uncertainty to the network training process because the training data distribution is changed, and proposed to switch off data augmentation at the end of the training stage to alleviate the empirical risk of optimization. Our method provides an alternative perspective that the risk is likely to be caused by the

noises of data augmentation and thus can be reduced by knowledge distillation. Moreover, the hyper-parameters in [15] (*e.g.*, when to switch off data augmentation) is difficult to tune. In training Wide-ResNet-28-10 [53] with AutoAugment on CIFAR-100, we follow the original paper to prevent data augmentation by adding 50 epochs to train the clean images only, but the baseline error rate (17.1%) is only reduced to 16.8%. In comparison, when knowledge distillation is added to these 50 epochs, the error rate is significantly reduced to 16.2%.

This work is also related to prior efforts that applied self-training to semi-supervised learning, *i.e.*, only a small portion of training data is labeled [42, 23, 46]. These methods often started with training a model on the labeled part, then used this model to 'guess' a pseudo label for each of the unlabeled samples, and finally updated the model using all data with either ground-truth or pseudo labels. This paper verifies the effectiveness of knowledge distillation in the fully-supervised setting in which augmented data can be noisy. Therefore, we draw the connection between exploring unseen data (data augmentation) and exploiting unlabeled data (semi-supervised learning), and reveal the potential of integrating AutoAugment and/or other hyper-parameter optimization methods to assist and improve semi-supervised learning.

4 Experiments

4.1 Results on the CIFAR-10/100 Datasets

Dataset and Settings. CIFAR-10/100 [21] contain tiny images with a resolution of 32×32. Both of them have 50K training and 10K testing images, uniformly distributed over 10 or 100 classes. They are two commonly used datasets for validating the basic properties of learning algorithms. Following the convention [4, 5], we train three types of networks, namely, wide ResNet (Wide-ResNet-28-10) [53], Shake-Shake (three variants with different feature dimensions) [10], and PyramidNet [12] with ShakeDrop regularization [49].

Knowledge Distillation Stabilizes AutoAugment. The core idea of our approach is to utilize knowledge distillation to restrain noises generated by severe transformations. This is expected to stabilize the training process of AutoAugment. To verify this, we start with training Wide-ResNet-28-10 on CIFAR-100. Note that the original augmentation space of AutoAugment involves two major kinds of transformations, namely, geometric or color-based transformations, on which AutoAugment as well as its variants limited the distortion magnitude of each transformation in a relatively small range so that the augmented images are mostly safe, *i.e.*, semantic information is largely preserved. In order to enhance the benefit brought by suppressing noises of aggressive augmentations, we design a new augment space in which the restriction in distortion magnitude is much weaker. To guarantee that large magnitudes lead to complete damage of semantic information, we only preserve a subset of geometric transformations (`shear-x`, `shear-y`, `translate-x`, `translate-y`) as well as `cutout`, and set 10 levels of distortion, so that $M = 0$ implies no augment, and $M = 10$ of any transformation destroys the entire image. Note that the range of M here is specifically designed

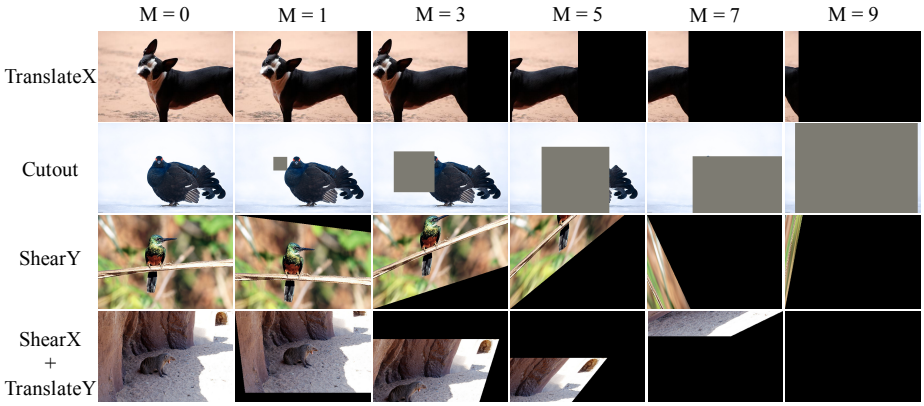


Fig. 3. Examples of transformations involved in our self-designed augment space. The distortion magnitude, M, is divided into 10 levels. The deformation introduced by transformations increases along with the magnitude. First three rows are examples of the deformation produced by each type of transformation with different magnitudes. The last row represents applying two consecutive transformations on a single image, which is the real case in our training scenario.

Model	**Distortion Magnitude,** M							
	0	1	2	3	4	5	6	7
RA	18.4	19.5	20.7	22.4	25.7	31.6	40.3	55.1
RA+KD	18.0	17.6	18.5	19.9	21.9	27.0	34.9	48.0
Gain	+0.4	+1.9	+2.2	+2.5	+3.8	+4.6	+5.4	+7.1

Table 1. Comparison between RandAugment with or without knowledege distillation in our self-designed augment space on CIFAR-100 based on Wide-ResNet-28-10. All numbers in the table are error rates (%). **M** indicates the distortion magnitude of each transformation. **RA** for RandAugment [5], and **KD** for knowledege distillation.

for the modified augment space, which is incomparable with the original definition of M in RandAugment (experimented in Section 4.2). Regarding knowledge distillation, we set $K = 3$ (computing KL-divergence between the distributions of top-3 classes, determined by the teacher model) for CIFAR-10 and $K = 5$ for CIFAR-100. The balancing coefficient, λ, and the softmax temperature, T, is set to be 1.0 and 2.0, respectively.

In this modified augment space, we experiment with the strategy of RandAugment [5] which controls the strength of augmentation by adjusting the distortion magnitude, M. For example, on the `translate-x` transformation, a magnitude of 3 allows the entire image to be shifted, to the left or right, by at most 30% of the visible field, and a magnitude of 10 enlarges the number into 100%, *i.e.*, the visible area totally disappears. More examples are shown in Figure 3. Note that RandAugment performs two consecutive transformations on

Dataset	Network	NA	AA	FAA	PBA	RA	Ours
CIFAR-10	Wide-ResNet-28-10	3.9	2.6	2.7	2.6	2.7	**2.4**
	Shake-Shake (26 2x32D)	3.6	2.5	2.5	2.5	–	**2.3**
	Shake-Shake (26 2x96D)	2.9	2.0	2.0	2.0	2.0	**1.8**
	Shake-Shake (26 2x112D)	2.8	**1.9**	**1.9**	2.0	–	**1.9**
	PyramidNet+ShakeDrop	2.7	**1.5**	1.7	**1.5**	**1.5**	**1.5**
CIFAR-100	Wide-ResNet-28-10	18.8	17.1	17.3	16.7	16.7	**16.2**
	Shake-Shake (26 2x96D)	17.1	14.3	14.6	15.3	–	**13.8**
	PyramidNet+ShakeDrop	14.0	10.7	11.7	10.9	–	**10.6**

Table 2. Comparison between our approach and other data augmentation methods on CIFAR-10 and CIFAR-100. The teacher for all networks is Wide-ResNet-28-10, except for PyramidNet+ShakeDrop with itself as teacher on CIFAR-100 (due to the huge performance gap). All numbers in the table are error rates (%). **NA** indicates no augmentation is used, **AA** for AutoAugment [4], **FAA** for fast AutoAugment [25], **PBA** for population-based augmentation [17], and **RA** for RandAugment [5].

each image, therefore, a magnitude of 8 is often enough to destroy all semantic contents. Hence, M is constrained within the range of 0–7 in our experiments.

Results of different distortion magnitudes are summarized in Table 1. With the increase of the magnitude, a larger portion of semantic information is expected to be removed from the training image. In this scenario, if we continue forcing the model to fit the ground-truth, one-hot supervision of each training sample, the deep network may get confused and 'under-fit' the training data. This causes consistent accuracy drop, especially in the modified augment space with only geometric transformations. Even when the full augment space is used (in which some transformations are not very sensitive to M), this factor persists and hinders the use of larger M values, and thus restricts the degree of freedom of AutoAugment.

Knowledge distillation offers an opportunity that each augmented image is checked beforehand, and a soft label is provided by a pre-trained teacher model to co-supervise the training process so that the deep network is not forced to fit the one-hot label. This is especially useful when the training image is contaminated by augmentation. As shown in Table 1, knowledge distillation provides consistent accuracy gain over RandAugment, as it slows down the accuracy drop with aggressive augmentation (the gain is larger as the distortion magnitude increases). More importantly, under a magnitude of $M = 1$, knowledge distillation produces an accuracy gain of 1.9%, assisting the RandAugment-only model with a deficit of 1.1% to surpass the baseline, claiming an advantage of 0.4%. This proves that the benefit mainly comes from the cooperation of RandAugment and knowledge distillation, not only from the auxiliary information provided by knowledge distillation itself [9, 1, 50].

Comparison with State-of-the-Arts. To make fair comparisons to the previous AutoAugment-based methods, we directly inherit the augmentation policies found on CIFAR by AutoAugment. In this full space, all transformations listed

in Section 3.1, not only the geometric transformations, can appear. Results are summarized in Table 2.

On CIFAR-10, our method outperforms other augmentation methods consistently, in particular, on top of smaller networks (*e.g.*, the error rates of Wide-ResNet-28-10 and two Shake-Shake models are reduced by 0.2%). For larger models, in particular PyramidNet with ShakeDrop regularization, the room of improvement on CIFAR-10 is very small, yet we can observe improvement on very large models on the more challenging CIFAR-100 and ImageNet datasets (see the next part for details).

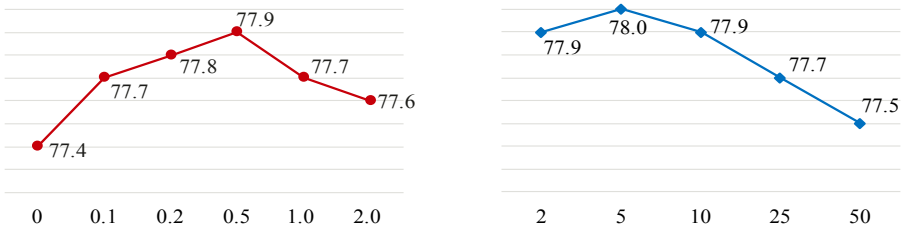
A side comment is that we have used the same teacher model (*i.e.*, Wide-ResNet-28-10, reporting a 2.6% error) which is relatively weak. We find this model can assist training much stronger students (*e.g.*, the Shake-Shake series, in which the error of the 2x96D model, 2.0%, is reduced to 1.8%). **In other words, weaker teachers can assist training strong students.** This delivers a complementary opinion to prior research which advocates for extracting 'dark knowledge' as some kind of auxiliary supervision [50] from stronger [16] or at least equally-powerful [9] teacher models, and further verifies the extra benefits brought by integrating knowledge distillation and AutoAugment together.

On CIFAR-100, we evaluate a similar set of network architectures, *i.e.*, Wide-ResNet-28-10, Shake-Shake (26 2x96D), and PyramidNet+ShakeDrop. As shown in Table 2, our results consistently outperform the previous state-of-the-arts. For example, on a relatively smaller Wide-ResNet-28-10, the error of AutoAugment decreases from 17.1% to 16.2% and significantly outperforms other methods, *e.g.*, PBA and RA. On Shake-Shake (26 2x96D), our approach also surpasses the previous best performance (14.3%) by a considerable margin of 0.5%. On pyramidNet with ShakeDrop, although the baseline accuracy is sufficiently high, knowledge distillation still brings a slight improvement (from 10.7% to 10.6%).

4.2 Results on the ImageNet Dataset

Dataset, Setting, and Implementation Details. ImageNet [6] is one of the largest visual recognition datasets which contains high-resolution images. We use the competition subset which has 1K classes, 1.3M training and 50K validation images. The number of images in each class is approximately the same for training data.

We build our baseline upon EfficientNet [41] and RandAugment [5]. EfficientNet contains a series of deep networks with different depths, widths and scales (*i.e.*, the spatial resolution at each layer). There are 9 variants of EfficientNet [45], named from B0 to B8. Equipped with RandAugment, EfficientNet-B7 reports a top-1 accuracy of 85.0% which is close to the state-of-the-art. We start with EfficientNet-B0 to investigate the impact of different knowledge distillation parameters on ImageNet, and finally compete with state-of-the-art results on EfficientNet-B4, EfficientNet-B7, and EfficientNet-B8.



(a) The impact of different λ values (b) The impact of different K values

Fig. 4. Training EfficientNet-B0 with different KD parameters. All numbers reported are top-1 accuracy (%). **Left**: The testing accuracy of different λ values, while K is set as 10. **Right**: The testing accuracy of different K values, while λ is set to be 0.5.

We follow the implementation details provided by the authors[1], and reproduce the training process using PyTorch. For EfficientNet-B0, it is trained through 500 epochs with an initial learning rate to be 0.256 and decayed by a factor of 0.97 every 2.4 epochs. We use the RMSProp optimizer with a decay factor of 0.9 and a momentum of 0.9. The batch-normalization decay factor is set to be 0.99 and the weight decay 10^{-5}. We use 32 GPUs (NVIDIA Tesla-V100) to train EfficientNet-B0/B4, and 256 GPUs for EfficientNet-B7/B8, respectively.

The Impact of Different Knowledge Distillation Parameters. We start with investigating the impact of λ and K, two important hyper-parameters of knowledge distillation. Note that we fix the softmax temperature, T, to be 10.0 in all ImageNet experiments. We perform experiments on EfficientNet-B0 with a moderate distortion magnitude of $M = 9$, which, as we have shown in the right-hand side of Figure 2, is a safe option on EfficientNet-B0. For λ, we set different values including 0.1, 0.2, 0.5, 1.0, and 2.0. For K, the optional values include 2, 5, 10, 25, and 50. To better evaluate the effect of each parameter, we fix one parameter value when changing the other.

Results are shown in Figure 4. It is clear that a moderate λ performs best. While setting λ with a small value, *e.g.*, 0.1, knowledge distillation is only expected to affect a small part of training samples. Yet, it obtains a 0.3% accuracy gain, implying that these samples, though rarely seen, can make the training process unstable. On the other hand, when λ is overly large, *e.g.*, knowledge distillation can dominate the training process and force the student model to have a very similar behavior to the teacher model, which limits its ability and harms classification performance.

Regarding K, we note that $K = 5$ achieves the best performance, indicating that on average, each class is connected to 4 other classes. This was also suggested in [50]. Yet, we find that setting $K = 2$ or $K = 10$ reports similar accuracy, but the performance gradually drops as K increases. This implies including too many classes for KL-divergence computation is harmful, because each training image, after augmented with a relatively small distortion magnitude, is not likely to be

[1] https://github.com/tensorflow/tpu/tree/master/models/official/efficientnet

connected to a large number of classes. However, to train more powerful models, larger distortion magnitudes need to be used and heavier ambiguity introduced. In this case, a larger K will be better, as we shall see in the next section.

Regardless of tuning hyper-parameters, we emphasize that all tested λ's, lying in the range of $[0.1, 2.0]$, and all tested K's, in $[2, 50]$, can bring positive effects on classification. This indicates that knowledge distillation is usually useful in training with augmented data. With the best setting, *i.e.*, a distortion magnitude of 9, a fixed K of 5, and $\lambda = 0.5$, we achieve a top-1 accuracy of 78.0% on EfficientNet-B0. This surpasses the accuracy of RandAugment (reproduced by us) and AdvProp [45] by margins of 0.6% and 0.4%, respectively.

Comparison to the State-of-the-Arts. To better evaluate the effectiveness of our approach, we further conduct experiments on more challenging large models, *i.e.*, EfficientNet-B4, EfficientNet-B7, and EfficientNet-B8. Given the fact that larger network is expected to over-fit more easily, for EfficienNet-B4 and EfficientNet-B7, we lift the magnitude of transformations on RandAugment from 9 in EfficientNet-B0 to 15 and 28, respectively. As discussed above, increasing the distortion magnitude brings more ambiguity to the training images so that each of them should be connected to more classes, and the knowledge distillation supervision should take a heavier weight. Hence, we increase K to 50 and λ to 2.0 in all experiments in this part.

Results are summarized in Table 3. By restraining the inevitable noises generated by RandAugment, our approach significantly boosts the baseline models. As shown in Table 3, the top-1 accuracy of EfficientNet-B4 is increased from 83.0% to 83.6%, and that of EfficientNet-B7 from 84.9% to 85.5%. The margin of 0.6% is considered significant in such powerful baselines. Both numbers surpass the current best, AdvProp [45], without using adversarial examples to assist training. Besides, when we simply double the training epochs of EfficientNet-B4, the top-1 accuracy is slightly improved from 83.0% to 83.2%, which is still much lower than 83.6% reported by applying another KD-guided training procedure.

Following AdvProp [45], we also move towards training EfficientNet-B8. The hyper-parameters remain the same as in training EfficientNet-B7. Due to GPU

Teacher Network	Student Network	AA	RA	RA†	AdvProp	Ours
EfficientNet-B0	EfficientNet-B0	77.3	–	77.4	77.6	**78.0**
EfficientNet-B4	EfficientNet-B4	83.0	–	83.0	83.3	**83.6**
EfficientNet-B7	EfficientNet-B7	84.5	85.0	84.9	85.2	**85.5**
EfficientNet-B7*	EfficientNet-B8	84.8	85.4	–	85.5	**85.7**

Table 3. Comparison between our approach and other data augmentation methods on ImageNet. All numbers in the table are top-1 accuracy (%). **AA** indicates AutoAugment [4] is used, **RA** for RandAugment [5], and **AdvProp** for Adversarial Propagation method [45]. **RA**† denotes the results of RandAugment produced by ourselves in PyTorch. EfficientNet-B7* denotes the student model in the penultimate row, which achieves a top-1 accuracy of 85.5%.

Method	Params	Extra Training Data	Top-1 (%)
ResNet-152 [14]	60M	–	77.8
Inception-v4 [37]	48M	–	80.0
ResNeXt-101 [47]	84M	–	80.9
SENet [18]	146M	–	82.7
AmoebaNet-C [32]	155M	–	83.5
GPipe [20]	557M	–	84.3
EfficientNet-B7 [5]	66M	–	85.0
EfficientNet-B8 [5]	88M	–	85.4
EfficientNet-L2 [41]	480M	–	85.5
AdvProp (EfficientNet-B8) [45]	88M	–	85.5
ResNeXt-101, Billion-scale [48]	193M	3.5B tagged images	84.8
FixRes ResNeXt-101, WSL [44]	829M	3.5B tagged images	86.4
Noisy Student (EfficientNet-L2) [46]	480M	300M unlabeled images	88.4
Ours (EfficientNet-B7 w/ KD)	66M	–	85.5
Ours (EfficientNet-B8 w/ KD)	88M	–	85.8

Table 4. Comparison to the state-of-the-arts on ImageNet. In the middie panel, we list three approaches with extra training data (a large number of weakly tagged or unlabeled images). Red and blue texts highlight the best results to date without and with extra training data, respectively.

memory limit, we use the best trained EfficientNet-B7 (with a 85.5% accuracy) as the teacher model. We report a top-1 accuracy of 85.7%, which sets the **new state-of-the-art** on the ImageNet dataset (without extra training data). With the test image size increased from 672 to 800, the accuracy is slightly improved to 85.8%. We show the comparison with previous best models in Table 4.

5 Conclusions

This paper integrates knowledge distillation into AutoAugment-based methods, and shows that the noises introduced by aggressive data augmentation policies can be largely alleviated by referring to a pre-trained teacher model. We adjust the computation of KL-divergence, so that the teacher and student models share similar probabilistic distributions over the top-ranked classes. Experiments show that our approach indeed suppresses noises introduced by data augmentation, and thus stabilizes the training process and enables more aggressive AutoAugment policies to be used. Our approach sets the new state-of-the-art, a 85.8% top-1 accuracy, on the ImageNet dataset (without extra training data).

Our research leaves several open problems. For example, it remains unclear whether useful information only exists in the top-ranked classes determined by the teacher model, and whether mimicking the class-level distribution is the optimal choice. Moreover, the balancing coefficient, λ, is a constant during training, which we believe there is room of improvement. We will continue investigating these topics in our future research.

References

1. Bagherinezhad, H., Horton, M., Rastegari, M., Farhadi, A.: Label refinery: Improving imagenet classification through label progression. arXiv preprint arXiv:1805.02641 (2018)
2. Brock, A., Lim, T., Ritchie, J.M., Weston, N.J.: Smash: One-shot model architecture search through hypernetworks. In: International Conference on Learning Representations (2018)
3. Chen, T., Goodfellow, I., Shlens, J.: Net2net: Accelerating learning via knowledge transfer. In: International Conference on Learning Representations (2016)
4. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation strategies from data. In: Computer Vision and Pattern Recognition (2019)
5. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. pp. 702–703 (2020)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition (2009)
7. DeVries, T., Taylor, G.W.: Improved regularization of convolutional neural networks with cutout. arXiv preprint arXiv:1708.04552 (2017)
8. Fang, H.S., Sun, J., Wang, R., Gou, M., Li, Y.L., Lu, C.: Instaboost: Boosting instance segmentation via probability map guided copy-pasting. In: International Conference on Computer Vision (2019)
9. Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., Anandkumar, A.: Born again neural networks. In: International Conference on Machine Learning (2018)
10. Gastaldi, X.: Shake-shake regularization. arXiv preprint arXiv:1705.07485 (2017)
11. Ghiasi, G., Lin, T.Y., Le, Q.V.: Nas-fpn: Learning scalable feature pyramid architecture for object detection. In: Computer Vision and Pattern Recognition (2019)
12. Han, D., Kim, J., Kim, J.: Deep pyramidal residual networks. In: Computer Vision and Pattern Recognition (2017)
13. Hataya, R., Zdenek, J., Yoshizoe, K., Nakayama, H.: Faster autoaugment: Learning augmentation strategies using backpropagation. arXiv preprint arXiv:1911.06987 (2019)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Computer Vision and Pattern Recognition (2016)
15. He, Z., Xie, L., Chen, X., Zhang, Y., Wang, Y., Tian, Q.: Data augmentation revisited: Rethinking the distribution gap between clean and augmented data. arXiv preprint arXiv:1909.09148 (2019)
16. Hinton, G., Vinyals, O., Dean, J.: Distilling the knowledge in a neural network. arXiv preprint arXiv:1503.02531 (2015)
17. Ho, D., Liang, E., Chen, X., Stoica, I., Abbeel, P.: Population based augmentation: Efficient learning of augmentation policy schedules. In: International Conference on Machine Learning (2019)
18. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Computer Vision and Pattern Recognition (2018)
19. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Computer Vision and Pattern Recognition (2017)
20. Huang, Y., Cheng, Y., Chen, D., Lee, H., Ngiam, J., Le, Q.V., Chen, Z.: Gpipe: Efficient training of giant neural networks using pipeline parallelism. In: Advances in Neural Information Processing Systems (2019)

21. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems (2012)
23. Laine, S., Aila, T.: Temporal ensembling for semi-supervised learning. In: International Conference on Learning Representations (2017)
24. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
25. Lim, S., Kim, I., Kim, T., Kim, C., Kim, S.: Fast autoaugment. In: Advances in Neural Information Processing Systems (2019)
26. Lin, C., Guo, M., Li, C., Yuan, X., Wu, W., Yan, J., Lin, D., Ouyang, W.: Online hyper-parameter learning for auto-augmentation strategy. In: International Conference on Computer Vision (2019)
27. Liu, C., Zoph, B., Neumann, M., Shlens, J., Hua, W., Li, L.J., Fei-Fei, L., Yuille, A., Huang, J., Murphy, K.: Progressive neural architecture search. In: European Conference on Computer Vision (2018)
28. Liu, H., Simonyan, K., Yang, Y.: Darts: Differentiable architecture search. In: International Conference on Learning Representations (2019)
29. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, Ł., Hinton, G.: Regularizing neural networks by penalizing confident output distributions. arXiv preprint arXiv:1701.06548 (2017)
30. Pham, H., Guan, M.Y., Zoph, B., Le, Q.V., Dean, J.: Efficient neural architecture search via parameter sharing. In: International Conference on Machine Learning (2018)
31. Real, E., Moore, S., Selle, A., Saxena, S., Suematsu, Y.L., Tan, J., Le, Q.V., Kurakin, A.: Large-scale evolution of image classifiers. In: International Conference on Machine Learning (2017)
32. Real, E., Aggarwal, A., Huang, Y., Le, Q.V.: Regularized evolution for image classifier architecture search. In: AAAI conference on Artificial Intelligence (2019)
33. Romero, A., Ballas, N., Kahou, S.E., Chassang, A., Gatta, C., Bengio, Y.: Fitnets: Hints for thin deep nets. arXiv preprint arXiv:1412.6550 (2014)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
35. Singh, K.K., Lee, Y.J.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In: International Conference on Computer Vision (2017)
36. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. Journal of Machine Learning Research **15**(1), 1929–1958 (2014)
37. Szegedy, C., Ioffe, S., Vanhoucke, V., A Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI conference on Artificial Intelligence (2017)
38. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: Computer Vision and Pattern Recognition (2015)
39. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)

40. Tan, M., Chen, B., Pang, R., Vasudevan, V., Sandler, M., Howard, A., Le, Q.V.: Mnasnet: Platform-aware neural architecture search for mobile. In: Computer Vision and Pattern Recognition (2019)
41. Tan, M., Le, Q.V.: Efficientnet: Rethinking model scaling for convolutional neural networks. In: International Conference on Machine Learning (2019)
42. Tarvainen, A., Valpola, H.: Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In: Advances in Neural Information Processing Systems (2017)
43. Thornton, C., Hutter, F., Hoos, H.H., Leyton-Brown, K.: Auto-weka: Combined selection and hyperparameter optimization of classification algorithms. In: SIGKDD International Conference on Knowledge Discovery and Data Mining (2013)
44. Touvron, H., Vedaldi, A., Douze, M., Jégou, H.: Fixing the train-test resolution discrepancy. In: Advances in Neural Information Processing Systems. pp. 8252–8262 (2019)
45. Xie, C., Tan, M., Gong, B., Wang, J., Yuille, A.L., Le, Q.V.: Adversarial examples improve image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 819–828 (2020)
46. Xie, Q., Luong, M.T., Hovy, E., Le, Q.V.: Self-training with noisy student improves imagenet classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 10687–10698 (2020)
47. Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: Computer Vision and Pattern Recognition (2017)
48. Yalniz, I.Z., Jegou, H., Chen, K., Paluri, M., Mahajan, D.: Billion-scale semi-supervised learning for image classification. arXiv preprint arXiv:1905.00546 (2019)
49. Yamada, Y., Iwamura, M., Akiba, T., Kise, K.: Shakedrop regularization for deep residual learning. IEEE Access **7**, 186126–186136 (2019)
50. Yang, C., Xie, L., Qiao, S., Yuille, A.L.: Training deep neural networks in generations: A more tolerant teacher educates better students. In: AAAI Conference on Artificial Intelligence (2019)
51. Yang, C., Xie, L., Su, C., Yuille, A.L.: Snapshot distillation: Teacher-student optimization in one generation. In: Computer Vision and Pattern Recognition (2019)
52. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: Cutmix: Regularization strategy to train strong classifiers with localizable features. In: International Conference on Computer Vision (2019)
53. Zagoruyko, S., Komodakis, N.: Wide residual networks. In: British Machine Vision Conference (2016)
54. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. In: International Conference on Learning Representations (2018)
55. Zhang, X., Wang, Q., Zhang, J., Zhong, Z.: Adversarial autoaugment. In: International Conference on Learning Representations (2020)
56. Zhang, Y., Xiang, T., Hospedales, T.M., Lu, H.: Deep mutual learning. In: Computer Vision and Pattern Recognition (2018)
57. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation. In: AAAI. pp. 13001–13008 (2020)
58. Zoph, B., Le, Q.V.: Neural architecture search with reinforcement learning. In: International Conference on Learning Representations (2017)
59. Zoph, B., Vasudevan, V., Shlens, J., Le, Q.V.: Learning transferable architectures for scalable image recognition. In: Computer Vision and Pattern Recognition (2018)