Improving Model Generalization by Agreement of Learned Representations from Data Augmentation

Rowel Atienza
University of the Philippines
Electrical and Electronics Engineering Institute, Diliman, 1101 Quezon City, Philippines
rowel@eee.upd.edu.ph

Abstract

Data augmentation reduces the generalization error by forcing a model to learn invariant representations given different transformations of the input image. In computer vision, on top of the standard image processing functions, data augmentation techniques based on regional dropout such as CutOut, MixUp, and CutMix and policy-based selection such as AutoAugment demonstrated state-of-the-art (SOTA) results. With an increasing number of data augmentation algorithms being proposed, the focus is always on optimizing the input-output mapping while not realizing that there might be an untapped value in the transformed images with the same label. We hypothesize that by forcing the representations of two transformations to agree, we can further reduce the model generalization error. We call our proposed method Agreement Maximization or simply AgMax. With this simple constraint applied during training, empirical results show that data augmentation algorithms can further improve the classification accuracy of ResNet50 on ImageNet by up to 1.5%, WideResNet40-2 on CIFAR10 by up to 0.7%, WideResNet40-2 on CIFAR100 by up to 1.6%, and LeNet5 on Speech Commands Dataset by up to 1.4%. Experimental results further show that unlike other regularization terms such as label smoothing, AgMax can take advantage of the data augmentation to consistently improve model generalization by a significant margin. On downstream tasks such as object detection and segmentation on PascalVOC and COCO, AgMax pre-trained models outperforms other data augmentation methods by as much as 1.0mAP (box) and 0.5mAP (mask). Code is available at https://github.com/roatienza/agmax.

1. Introduction

To achieve state-of-the-art (SOTA) performance, data augmentation plays a crucial role in both supervised and self-supervised model training. In computer vision, image processing functions such as rotation, translation, cropping, flipping, and color distortion improve model generalization. In recent years, a strong interest in new data augmentation techniques has emerged because of the significant improvement in model performance compared to baseline scores.

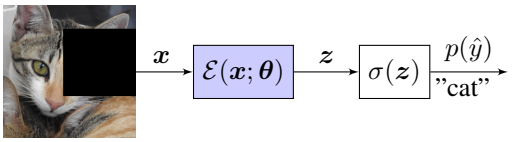

Figure 1. In supervised learning, a model $\mathcal{E}$ is trained to find the optimal parameters $\boldsymbol{\theta}^*$. Input data augmentation improves the performance by forcing $\mathcal{E}$ to learn invariant representations $\boldsymbol{z}$ under image transformation such as removing a random square region as done in CutOut. $\boldsymbol{x}$ is a labelled positive sample.

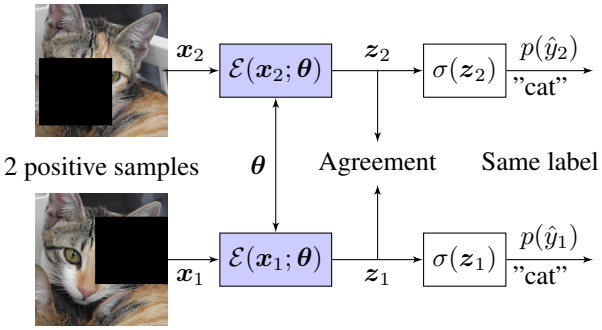

Figure 2. In supervised learning with *AgMax*, we impose an additional constraint that representations $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ must also agree. The two parallel models are just one and the same and share the same set of parameters $\boldsymbol{\theta}$. $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are 2 positive samples with the same label.

Instead of just applying random image processing operations, policy-based methods such as AutoAugment (AA) [5], FastAugment [30], RandAugment [6], Adversarial AutoAugment [53], and PBA [22] carefully select a recipe of image operations to generate new input data to minimize

Data Augmentation	Baseline	with Label Smoothing	with AgMax
Standard	76.4 ± 0.1	76.8 ± 0.1	76.9 ± 0.1
CutOut[7]	76.2 ± 0.0	76.5 ± 0.1	77.1 ± 0.0
MixUp[51]	76.5 ± 0.1	76.7 ± 0.1	$\mathbf{77.6} \pm 0.1$
CutMix[49]	76.3 ± 0.0	76.4 ± 0.1	77.4 ± 0.0
AA[5]	76.2 ± 0.1	76.2 ± 0.1	77.1 ± 0.1
CutOut+AA[5]	75.7 ± 0.1	75.7 ± 0.1	76.6 ± 0.1
MixUp+AA	75.9 ± 0.0	76.5 ± 0.1	77.1 ± 0.1
CutMix+AA	75.5 ± 0.1	75.5 ± 0.1	77.0 ± 0.1

Table 1. Top-1% accuracy of ResNet50 trained for 90 epochs on ImageNet using different data augmentation methods with Label Smoothing or AgMax. Standard data augmentation is random horizontal flipping, color jitter and lighting.

model overfitting. Regional dropouts or techniques based on direct image alteration such as CutOut [7], RICAP [42], CutMix [49], GridMask [4] and MixUp [51] improve model performance by forcing the model to learn invariant representations.

With an increasing number of data augmentation algorithms being introduced, as shown in Figure 1 the focus is always on optimizing the input-output mapping. We hypothesize that there might be an untapped value between representations of transformed inputs with the same label. Basic intuition tells us that under different transformations such as in Figure 2, two inputs with the same label should agree on which representations that a model learns. These two inputs are called **positive samples** since they have the same label but two different transformations. A classifier receiving two positive samples of a *cat* must learn to extract the *minimum common set of representations* such as the presence of whiskers, fur, sharp eyes, short nose, etc. We call this function *Agreement*. In this paper, we use mutual information (MI) to estimate *Agreement*. We performed ablation studies to demonstrate that other agreement functions such as MSE, KL-divergence and cross-entropy (CE) are also effective.

Using a common evaluation protocol, experimental results indicate that our proposed method *AgMax* improves the performance of almost all model-dataset configurations. On ResNet50 trained on ImageNet for 90 epochs, as shown in Table 1, *AgMax* consistently outperforms Label Smoothing [41] especially under heavy data augmentation. In a bigger evaluation landscape, the results in Tables 2 and 3 demonstrate the consistent improvement in generalization for different models and datasets due to *AgMax*. On downstream tasks such as object detection and segmentation, a ResNet50 model pre-trained with *AgMax* outperforms its counterpart pre-trained model by as much as 1.0mAP on *bounding box* and 0.5mAP on *segmentation mask*.

2. Related Work

To achieve state-of-the-art (SOTA) performance, data augmentation plays a crucial role in both supervised and self-supervised model training. Data augmentation belongs to a bigger field of study called regularization. The objective of regularization is to improve model generalization by modifying the network structure during training, augmenting the train dataset, modifying the loss function or modifying the model training algorithm. For example, dropout [40] randomly drops neural network units during training to mimic data and network perturbations. As a result, a model improves its test performance. In deep CNNs, instead of dropping feature maps, noise injection or substitution such as Stochastic Depth [23], Shake-Shake [10], DropBlock [12], DropPath [27], SpatialDropout [44] and Shake-Drop [48] are used. Related to dropout is the regional dropout. Instead of dropping neural network units, a certain region of the input is removed, mixed or blended. In effect, regional dropout augments the training dataset by exposing a model to extreme input data transformations. CutOut [7], RICAP [42], CutMix [49], GridMask [4] and MixUp [51] belong to this category.

Before the regional dropout methods were proposed, data augmentation was achieved by basic input data transformations. In computer vision, padding, random cropping, translation, rotation, horizontal flipping and color distortion are commonly used. Recently, these standard data augmentation techniques have been supplanted by a more structured learned policy in order to arrive at an optimal recipe of data transformation functions. In computer vision, AutoAugment [5], FastAugment [30], Adversarial AutoAugment [53], RandAugment [6] and PBA [22] have been proposed. Among the available data augmentation methods, regional dropouts, policy-based, and gradient augmentation have demonstrated state-of-the-art results. Improving model generalization can also be achieved by modifying the loss function. Label smoothing [41] and weight decay [15] belong to this category.

In this paper, the idea is to take advantage of the huge amount of data generated by augmentation methods. If an image undergoes data augmentation to produce 2 new images, the resulting representations a model learns must agree since both inputs belong to the same category. This idea bears resemblance to representations matching in self-supervised learning such as BYOL [14] and DINO [2] where the predictions of teacher(target) and student(online) networks on 2 positive samples (2 views of the same image) are reinforced by similarity learning. The similarity function could be MSE in BYOL or cross-entropy (CE) function in DINO. The teacher and student networks could be similar in architecture but have different sets of parameters. The teacher network parameters are exponential moving average of the online parameters. The main difference of *Ag-*

Data Augmentation	Year	CIFAR10[25] WideResNet [50] 200 epochs 40-2	CIFAR10[25] WideResNet [50] 200 epochs 28-10	CIFAR100 WideResNet [50] 200 epochs 40-2	CIFAR100 WideResNet [50] 200 epochs 28-10	ImageNet [38] ResNet50 [17] 90 epochs Top-1	ImageNet [38] ResNet50 [17] 90 epochs Top-5	ImageNet [38] ResNet50 [17] 270 epochs Top-1	ImageNet [38] ResNet50 [17] 270 epochs Top-5
Published Results									
Standard	-	94.7	96.1	74.0	81.2	76.2	92.9	76.3	93.1
CutOut[7]	2017	95.9	96.9	74.8	81.6	-	-	77.1	93.3
MixUp[51]	2018	-	97.3	-	82.5	76.7	93.4	77.9	93.9
CutMix[49]	2019	-	-	-	-	-	-	78.6	94.1
AA[5]	2019	-	-	-	-	-	-	77.6	93.8
CutOut+AA[5]	2019	96.3	97.4	79.3	82.9	-	-	-	-
without AgMax (Reproduced Results)									
Standard	-	95.1	96.2	76.9	81.3	76.4	93.2	76.8	93.3
CutOut[7]	2017	96.2	97.1	78.4	82.5	76.2	93.1	77.5	93.6
MixUp[51]	2018	95.8	97.1	78.3	82.8	76.5	93.3	78.2	93.9
CutMix[49]	2019	96.2	97.3	79.4	83.8	76.3	93.2	78.7	94.2
AA[5]	2019	95.9	96.9	78.4	82.8	76.2	93.1	77.6	93.6
CutOut+AA[5]	2019	96.4	97.5	80.0	83.8	75.7	92.8	77.9	93.7
MixUp+AA	-	96.0	97.4	79.1	84.1	75.9	92.9	78.3	94.0
CutMix+AA	-	96.4	97.4	80.1	85.0	75.5	92.7	78.5	94.1
with AgMax									
Standard	-	95.6(0.5)	96.4(0.2)	77.4(0.5)	81.7(0.4)	76.9(0.5)	93.5(0.3)	77.2(0.4)	93.6(**0.3**)
CutOut	-	96.6(0.4)	97.3(0.2)	79.2(0.8)	82.6(0.1)	77.1(0.9)	93.6(0.5)	77.6(0.1)	93.8(0.2)
MixUp	-	96.3(0.5)	97.5(0.4)	79.5(1.2)	82.9(0.1)	**77.6**(1.1)	93.7(0.4)	78.4(0.2)	94.1(0.2)
CutMix	-	96.7(0.5)	97.7(0.4)	80.0(0.6)	84.0(0.2)	77.4(1.1)	**93.9**(0.7)	79.0(0.3)	94.2(0.0)
AA	-	96.4(0.5)	97.4(**0.5**)	79.2(0.8)	83.0(0.2)	77.1(0.9)	93.4(0.3)	77.7(0.1)	93.8(0.2)
CutOut+AA	-	**97.1(0.7)**	97.8(0.3)	81.1(1.1)	84.0(0.2)	76.6(0.9)	93.4(0.6)	78.2(0.3)	94.0(**0.3**)
MixUp+AA	-	96.6(0.6)	**97.9(0.5)**	80.7(**1.6**)	84.8(**0.7**)	77.1(1.2)	93.5(0.6)	78.6(0.3)	94.2(0.2)
CutMix+AA	-	96.8(0.4)	97.8(0.4)	**81.3**(1.2)	**85.3**(0.3)	77.0(**1.5**)	93.5(**0.8**)	**79.1(0.6)**	**94.4(0.3)**

Table 2. Evaluation landscape showing model accuracy of different data augmentation algorithms with and without *AgMax*. Underscore is the best performing configuration without *AgMax*. Bold is the best performing method for all configurations. The absolute percentage increase in accuracy due to *AgMax* is enclosed in parentheses. Standard data augmentation algorithm is defined in the Experimental Results section.

Max is it uses one network for both predictions. Instead of maximizing a similarity function, *AgMax* maximizes the agreement using mutual information.

In this paper, we validate our hypothesis on 4 commonly used data augmentation algorithms: 1 policy-based approach and 3 regional dropout techniques. We chose AutoAugment for the policy-based data augmentation given that its policy is publicly available. FastAugment, Adversarial AutoAugment, and RandAugment are built on top of the key ideas of AutoAugment. For regional dropout, we used CutOut, MixUp, and CutMix. These methods have gained mainstream use and achieved state-of-the-art results. Furthermore, their code implementations are publicly available for reproducibility.

3. Improving Generalization by Agreement

With reference to Figure 2, we hypothesize that the representations of two images derived by applying a data augmentation method on an image must agree for a model to further improve its classification accuracy. In this paper, we propose that the *Agreement* is the amount of shared information between the two views of the same image. Therefore, on top of the classification loss function, maximizing the mutual information between the two representations could improve the model generalization. The total loss function can be expressed as:

$$\mathcal{L} = \mathcal{L}_{CE} + \lambda \mathcal{L}_{MI}. \tag{1}$$

λ is the weight of the MI loss function.

For the case of discrete random variables such as in image classification, mutual information is expressed as:

$$I(\boldsymbol{z}_1; \boldsymbol{z}_2) = \sum_{\boldsymbol{z}_1, \boldsymbol{z}_2} P(\boldsymbol{z}_1, \boldsymbol{z}_2) \log \frac{P(\boldsymbol{z}_1, \boldsymbol{z}_2)}{P(\boldsymbol{z}_1) P(\boldsymbol{z}_2)}. \tag{2}$$

In other words, MI is the KL-divergence between the joint and product of marginal probabilities of $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$.

In recent years, several neural MI estimators have been proposed [1, 45, 21, 24, 33]. Invariant Information Clus-

Data Augmentation	Speech Commands [47] LeNet5 [28] 30 epochs test	val	VGG11 [39] test	val
Published Results				
Standard	89.7	90.2	95.4	95.0
CutOut[7]	-	-	-	-
MixUp[51]	89.2	89.9	96.6	96.1
CutMix[49]	-	-	-	-
without AgMax (Reproduced Results)				
Standard	89.9	90.0	96.3	96.0
CutOut[7]	89.0	89.2	96.5	96.1
MixUp[51]	89.4	89.4	96.5	96.2
CutMix[49]	87.1	87.5	96.4	96.2
with AgMax				
Standard	90.2(0.3)	**90.0**(0.0)	96.4(0.1)	96.1(0.1)
CutOut	**90.4**(1.4)	**90.0**(0.8)	96.5(0.0)	96.1(0.0)
MixUp	89.4(0.0)	89.6(0.2)	**96.8(0.3)**	96.3(0.1)
CutMix	88.8(**1.7**)	89.3(**1.8**)	96.7(**0.3**)	**96.4(0.2)**

Table 3. Evaluation landscape showing Top-1% model accuracy of different data augmentation configurations on Speech Commands Dataset. AutoAugment is not included since there is no publicly available policy for Speech Commands Dataset.

tering (IIC) [24] proposed a method to estimate Equation 2.

For a given dataset or batch of size n, the joint probability matrix $\boldsymbol{P} \in \mathbb{R}^{C\times C}$ can be computed as:

$$\boldsymbol{P} = \frac{1}{n}\sum_{i=1}^{n} \Phi(\boldsymbol{x}_{1,i}) \cdot \Phi(\boldsymbol{x}_{2,i})^{\top}, \tag{3}$$

where $\boldsymbol{x}_{1,i}$ and $\boldsymbol{x}_{2,i}$ are two transformed versions of the same image $\boldsymbol{x}_i$. $\Phi(\boldsymbol{x}) = \sigma(\mathcal{E}(\boldsymbol{x})) = softmax(\boldsymbol{z}) \in [0,1]^C$. This can be interpreted as the distribution of z over C classes formally given as $P(z = c|\boldsymbol{x}, \boldsymbol{\theta})) = \Phi_c(\boldsymbol{x})$. The marginal distributions $\boldsymbol{P}_{c_1} = P(z_1 = c_1)$ and $\boldsymbol{P}_{c_2} = P(z_2 = c_2)$ can be obtained by summing the rows and columns of $\boldsymbol{P}$ respectively. Each element of $\boldsymbol{P}$ is the joint probability $\boldsymbol{P}_{c_1c_2} = P(z_1 = c_1, z_2 = c_2)$. Since $\boldsymbol{P}_{c_1c_2} = \boldsymbol{P}_{c_2c_1}$, the matrix $\boldsymbol{P}$ must be symmetric. Ensuring a symmetric $\boldsymbol{P}$ is done by $\boldsymbol{P} = \frac{\boldsymbol{P}+\boldsymbol{P}^{\top}}{2}$.

Using the joint and marginal probabilities, the mutual information loss is computed as:

$$\mathcal{L}_{MI} = -I(\boldsymbol{z}_1; \boldsymbol{z}_2) = -\sum_{c_1=1}^{C}\sum_{c_2=1}^{C} \boldsymbol{P}_{c_1c_2} \ln \frac{\boldsymbol{P}_{c_1c_2}}{\boldsymbol{P}_{c_1} \cdot \boldsymbol{P}_{c_2}}. \tag{4}$$

A 2-layer MLP network can also be used to estimate the joint distribution. The MLP network is trained using the objective functions: $p(\boldsymbol{z}_1, \boldsymbol{z}_2) \rightarrow p(\boldsymbol{z}_1, \boldsymbol{z}_1)$ and $p(\boldsymbol{z}_1, \boldsymbol{z}_2) \rightarrow p(\boldsymbol{z}_2, \boldsymbol{z}_2)$ since both features refer to the same class and so is the joint distribution.

Note that to estimate the MI loss, only pairs of positive samples or one-to-one mapping is needed. Therefore, *AgMax* works even for small batch sizes. This is different from constrastive learning that requires a positive and many negative samples or one-to-many mapping. Contrastive learning needs large batch sizes (*e.g.* 4,096 and up) to work. This has a huge negative implication on GPU memory requirements.

3.1. Agreement by Mutual Information

In this section, we **attempt** to find a possible explanation on why MI provides a good *Agreement* function using the Maximum a Posteriori (MAP) principle:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log p(\boldsymbol{\theta}|\mathcal{D}) = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \log p(\boldsymbol{\theta}|\boldsymbol{x}, \boldsymbol{y}). \tag{5}$$

When applied to deep neural networks (DNNs) as shown in Figure 1, $\boldsymbol{\theta}$ is the model parameters while $\boldsymbol{\theta}^*$ represents the maximal point estimate for a given dataset $\mathcal{D} = \{\boldsymbol{x}, \boldsymbol{y}\}$.

Using Bayes' Theorem and with the constant term dropped, the conditional probability in Equation 5 can be rewritten as:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) + \log p(\boldsymbol{x}|\boldsymbol{\theta})p(\boldsymbol{\theta})\right]. \tag{6}$$

In a given model, a backbone network encodes the input into a latent variable, $\boldsymbol{z} = \mathcal{E}(\boldsymbol{x}; \boldsymbol{\theta}_1)$. A decoder decides what output is generated, $\hat{\boldsymbol{y}} = \mathcal{D}(\boldsymbol{z}; \boldsymbol{\theta}_2)$. Collectively, $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2\}$. If $\boldsymbol{z}$ is taken from the last layer before the non-parametric softmax prediction of $\hat{\boldsymbol{y}} = \sigma(\boldsymbol{z})$, then $\boldsymbol{\theta} = \{\boldsymbol{\theta}_1, \varnothing\}$ and $\hat{\boldsymbol{y}}$ is a good proxy of $\boldsymbol{z}$. During supervised training, the goal is to estimate the empirical distribution $p(\boldsymbol{y}|\boldsymbol{x})$ using a parameterized distribution $p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{\theta})$ as modelled by the encoder-decoder. This is done by minimizing a distance function such as the Kullback-Leibler divergence function $D_{KL}(p(\boldsymbol{y}|\boldsymbol{x}) \parallel p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{\theta}))$. In supervised classification problems, this is equivalent to minimizing the categorical cross-entropy loss function, $\mathcal{L}_{CE} = -\mathbb{E}_{p(\boldsymbol{y}|\boldsymbol{x})} \log p(\hat{\boldsymbol{y}}|\boldsymbol{x}, \boldsymbol{\theta})$.

As shown in Figure 2, given two positive samples of $\boldsymbol{x}$, we can reformulate MAP as:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[\frac{1}{2}\sum_{i=1}^{2} \mathbb{E} \log p(\boldsymbol{y}|\boldsymbol{x}_i, \boldsymbol{\theta}) + \frac{1}{2}\mathbb{E}\Big[\log \frac{p(\boldsymbol{z}_1, \boldsymbol{z}_2|\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{\theta})}{p(\boldsymbol{z}_1|\boldsymbol{x}_1, \boldsymbol{\theta})p(\boldsymbol{z}_2|\boldsymbol{x}_2, \boldsymbol{\theta})}\Big] + \frac{1}{2}\mathbb{E} \log p(\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{\theta})\right], \tag{7}$$

since Equation 6 can be rewritten as:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[\log p(\boldsymbol{y}|\boldsymbol{x}, \boldsymbol{\theta}) + \log p(\boldsymbol{z}, \boldsymbol{x}, \boldsymbol{\theta}) - \log p(\boldsymbol{z}|\boldsymbol{x}, \boldsymbol{\theta})\right]. \quad (8)$$

Given two data augmentations for a given input:

$$\boldsymbol{\theta}^* = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \left[\frac{1}{2}\sum_{i=1}^{2} \log p(\boldsymbol{y}|\boldsymbol{x}_i, \boldsymbol{\theta}) + \frac{1}{2}\log p(\boldsymbol{z}_1, \boldsymbol{x}_1, \boldsymbol{\theta}) p(\boldsymbol{z}_2, \boldsymbol{x}_2, \boldsymbol{\theta}) - \frac{1}{2}\sum_{i=1}^{2} \log p(\boldsymbol{z}_i|\boldsymbol{x}_i, \boldsymbol{\theta})\right], \quad (9)$$

Assuming independence, the second term in Equation 9 can be expressed as:

$$\log p(\boldsymbol{z}_1, \boldsymbol{x}_1, \boldsymbol{\theta}) p(\boldsymbol{z}_2, \boldsymbol{x}_2, \boldsymbol{\theta}) = \log p(\boldsymbol{z}_1, \boldsymbol{z}_2, \boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{\theta}). \quad (10)$$

With dataset sampling, Equation 9 can be rewritten as Equation 7. The second term in Equation 7 resembles the mutual information (MI): $I(\boldsymbol{z}_1; \boldsymbol{z}_2|\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{\theta})$. Maximizing the MI between the representations of $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ regularizes the choice of model parameters $\boldsymbol{\theta}$. In Figure 2 for example, the network is encouraged to find the representations of two views of the same *cat*, such that the shared information between $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$ is maximized. This could be finding the common features such as presence of whiskers, fur, sharp eyes, short nose, etc.

The third term in Equation 7 can be rewritten as $\frac{1}{2}\mathbb{E}\log p(\boldsymbol{\theta})$ since the model parameters can be assumed to be independent from the input distribution. This term can be represented by $L2$ weight regularization. The first term of Equation 7 can be optimized by the standard cross entropy loss function as discussed earlier in this section. Therefore, the total loss function can be written as Equation 1.

4. Experimental Results

To validate our hypothesis, we first reproduced different data augmentation results. Then, we applied *AgMax* to verify if the model generalization will improve.

To arrive at a fair comparison, we looked for the common evaluation protocols among the data augmentation algorithms under study starting with datasets and encoders or backbone networks. CIFAR10, CIFAR100 and ImageNet datasets are commonly used. We included Speech Commands Dataset as evaluated by MixUp. Except for CutOut, all methods used ResNet50 as the backbone network on ImageNet. We included WideResNet28-10 and WideResNet40-2 as used by AutoAugment, RandAugment, and MixUp on CIFAR10, and CIFAR100. Lastly, we

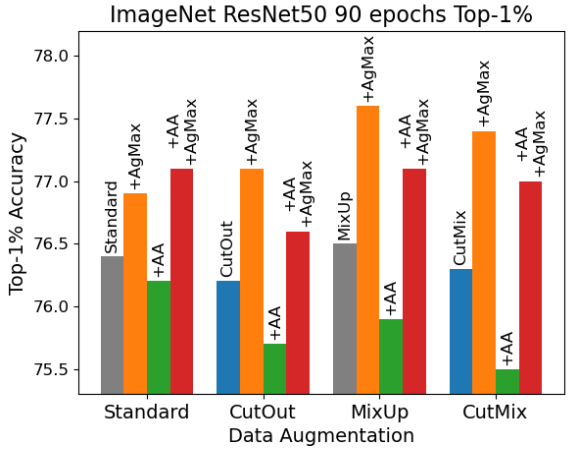

Figure 3. Top-1% accuracy of different regularizer configurations on ImageNet using ResNet50 trained for 90 epochs.

trained VGG11 and LeNet5 on Speech Commands Dataset as done in MixUp.

However, an analysis of published experiments revealed that it is difficult to make a fair comparison of scores produced by different data augmentation algorithms due to a lack of consistent evaluation protocol. For example, on ImageNet dataset, the ResNet50 model was trained for epochs ep=90 and 200, lr=0.4, bs=1, 024 in MixUp, and ep=300, lr=0.1, bs=256 in CutMix. It leads to an unfair comparison since the performance of ResNet varies with the number of epochs, batch size, and learning rate settings. To address these issues, we formulated a common training condition. Then, we reproduced the reported scores. In the following subsections, we discuss the details of the uniform experimental setup that we used for each model and dataset. Whenever possible, we implemented the settings in the published literature or official code implementations.

A further examination of published experiments showed that policy-based methods are seldom evaluated with complementary regional dropout algorithms. For example, AutoAugment has demonstrated that it can achieve better results with CutOut but its use with other regional dropout algorithms was not fully exploited. In the Published Results section of Table 2, only 58% of the evaluation space has data. In our experiments, combinations of complementary data augmentation algorithms were also examined. This enabled us to see the big picture of the evaluation space.

To arrive at the results in Tables 2 and 3, all models were trained from scratch using random seeds for at least 3 times with and without *AgMax*. The best test scores in each run were averaged for the evaluation reporting. We used the default parameter initialization in PyTorch [34] but we observed improvement in performance if a higher entropy Gaussian distribution is used in *AgMax*. For the MI loss function, $\lambda = -1$ in our experiments.

From Tables 2 and 3 and Figure 3, we make the following general observations:

1. *AgMax* consistently improves the performance of all data augmentation algorithms either as a standalone or in combination with other methods. For ResNet50 (90 epochs ImageNet in Figure 3) and LeNet5 (30 epochs Speech Commands in Table 3), only *AgMax* achieves significant generalization improvements.

2. There is no data augmentation algorithm, separately or in combination with AutoAugment and/or *AgMax* that can outperform all other methods in all datasets, models, and training conditions. This means that there is no single superior data augmentation method among the techniques that we evaluated.

3. AutoAugment improves the performance of baseline and regional dropout algorithms. Exceptions are on ResNet50 at 90 epochs and on CutMix at 270 epochs. Both configurations were not explored in their original papers.

4. Similar to policy-based methods, *AgMax* can be applied as an add-on regularizer to improve model generalization. This simplifies the overall optimization process.

4.1. CIFAR10 and CIFAR100

Both CIFAR10 and CIFAR100 datasets [25] have 60,000 real-world color images of size 32×32 pixels. Both datasets have train-test split of 50,000-10,000. CIFAR10 has 10 classes while CIFAR100 is made of 100 classes.

All regularization methods were evaluated after training WideResNet28-10 and WideResNet40-2 models [50] with an initial learning rate of 0.1 for 200 epochs using cosine learning rate decay, batch size of 128, and SGD optimizer with weight decay of $5e^{-4}$. The standard data augmentation is made of random cropping of 32×32 pixels with padding size of 4 pixels, random horizontal flipping, and normalization. CutOut size is 16×16 pixels. MixUp has α=1. CutMix probability is 0.5 with Beta distribution of α=1.

Table 2 shows that 2 out of 4 model and dataset configurations, *CutMix+AA+AgMax* is the top performing. The rest is split between *CutOut+AA+AgMax* and *MixUp+AA+AgMax*.

4.2. ImageNet

The ILSVRC ImageNet dataset [38] has 1.2M real-world color images made of 1,000 classes for training. The validation set has 50,000 color images. The standard image processing includes resizing, cropping to 224×224 pixels, random horizontal flipping, color jitter, lighting, and normalization.

All data augmentation methods were evaluated after training a ResNet50 network for 90 and 270 epochs. The initial learning rate is 0.1. The optimizer is SGD with a weight decay of $1e^{-4}$. The learning rate is reduced by a factor of 0.1 at 30, 60, and 80 epochs during the 90 epochs training and 75, 150, and 225 epochs during the 270 epochs training. Batch size is 256. CutOut size is 112×112 pixels as used in CutMix paper. MixUp has α=0.2. CutMix probability is 1.0 with Beta distribution of α=1.

Table 2 and Figure 3 show that when ResNet50 is trained for 90 epochs, *MixUp+AgMax* is the best performing method while *CutMix+AA+AgMax* is the top performing algorithm for 270 epochs. When only one regularizer is combined with the standard data augmentation, only *AgMax* has a significant positive gain in accuracy and therefore has the highest performance in 90 epochs.

4.3. Speech Commands Dataset

Google Speech Commands Dataset [47] contains 64,727 30-class utterances from 1,881 speakers. Each single-word utterance such as *yes, no, up,* or *down* is about one-second long. The standard audio signal preprocessing include random amplitude, pitch, and speed adjustment, stretching, time shifting, and addition of background noise. The audio signals are converted into 32×32 mel spectrogram input data.

Table 3 data augmentation methods were evaluated after training LeNet5 [28] and VGG11 [39] for 30 epochs. The initial learning rate is 0.003 and adjusted by a factor of 0.1 after every 10 epochs for LeNet5 and 15 epochs for VGG11. The optimizer is Adam with a weight decay of $5e^{-4}$ for LeNet5 and $1e^{-4}$ for VGG11. The settings in CIFAR10/100 for CutOut, MixUp, and CutMix are used.

Table 3 shows that *CutOut+AgMax* is the best performing algorithm for LeNet5 and a tie between *MixUp+AgMax* and *CutMix+AgMax* for VGG11. Except for *AgMax*, we observed that when used as a standalone regularizer all data augmentation algorithms fail to improve the generalization of LeNet5.

4.4. Robustness

Robustness has been increasing in importance as we deploy deep learning models on safety-critical applications. **Although the regularization methods that we evaluated are not consciously optimized for robustness**, it is worth knowing how our different configurations perform under data corruption and adversarial attack. We evaluated our ResNet50 models trained for 270 epochs using a comprehensive data corruption suite called ImageNet-C [19]. For the adversarial white-box attack, we subjected our trained models under Fast Gradient Sign Method (FGSM) [13].

The mean Corruption Error (mCE) in Table 4 is normalized with respect to AlexNet [26] performance on ImageNet as proposed by Hendrycks *et al*. [19]. In Table 4, the configurations that are most robust against data corruption have one method in common - MixUp. This confirms the MixUp robustness study by Zhang *et al*. [52]. In many cases, AutoAugment and *AgMax* further improve

Data Augmentation	mCE↓	Noise↓ Gauss	Shot	Impulse	Blur↓ Defocus	Glass	Motion	Zoom	Weather↓ Snow	Frost	Fog	Bright	Digital↓ Contrast	Elastic	Pixel	JPEG
without AgMax																
Standard	74.8	71.8	73.3	76.5	79.2	91.0	82.3	80.9	74.3	73.0	61.6	57.8	65.2	89.1	71.7	73.9
CutOut	75.2	74.3	76.4	80.8	77.3	91.2	79.1	79.8	75.5	74.0	63.3	57.1	64.7	87.9	73.4	73.8
MixUp	69.9	65.0	68.7	70.1	76.8	90.2	78.7	77.6	67.6	62.7	54.9	53.9	54.5	**87.6**	68.7	71.9
CutMix	75.0	74.5	76.8	81.8	79.0	92.3	81.2	78.2	73.5	73.8	62.0	55.9	64.8	88.9	68.9	73.9
AA	72.6	66.6	67.1	71.5	77.6	88.8	78.1	83.2	72.9	73.2	60.3	54.2	60.2	91.0	71.9	72.5
CutOut+AA	72.2	67.6	68.6	72.0	75.4	90.0	78.6	81.8	73.2	74.0	59.3	54.2	58.6	92.6	68.3	68.5
MixUp+AA	67.6	59.8	60.6	63.8	**74.9**	**86.9**	**74.9**	78.2	66.4	64.7	53.3	52.3	53.3	88.6	69.0	**67.1**
CutMix+AA	72.2	66.4	67.0	72.5	76.7	91.5	77.5	78.1	73.3	72.8	58.8	53.3	60.1	91.1	71.7	72.3
with AgMax																
Standard	74.9	72.4	74.6	77.9	78.5	89.9	81.3	82.4	75.0	73.3	62.3	56.3	64.3	89.7	70.8	74.3
CutOut	74.9	72.7	75.1	78.6	78.4	91.8	79.5	79.9	74.5	74.5	61.7	55.8	63.6	89.5	74.6	73.6
MixUp	69.9	65.0	68.7	69.0	76.8	90.4	77.8	**77.3**	67.8	61.2	54.9	54.4	56.3	89.0	**67.9**	72.1
CutMix	75.3	74.5	76.7	80.4	78.2	91.7	79.0	**77.3**	74.4	74.2	62.0	56.2	64.8	89.6	74.9	76.1
AA	73.7	68.7	69.6	73.9	79.3	88.7	79.4	83.5	73.8	73.3	62.2	55.5	59.4	92.5	72.0	74.3
CutOut+AA	73.3	68.4	69.0	74.6	77.1	90.6	78.2	83.6	75.0	74.5	61.3	54.1	58.8	92.5	71.7	70.8
MixUp+AA	**67.1**	**57.2**	**58.5**	**60.8**	75.8	89.6	76.2	78.7	**64.4**	**60.2**	**52.1**	**51.9**	**52.5**	90.4	69.1	69.8
CutMix+AA	72.9	70.8	71.5	75.4	75.8	91.9	77.8	78.6	72.2	72.9	58.1	52.8	58.9	90.0	74.0	72.9

Table 4. Evaluation landscape showing corruption robustness of ResNet50 model trained for 270 epochs using different regularizer configurations. mCE is Top-1% mean Corruption Error.

Method	Year	Stand-alone?	mCE↓	Clean Error↓
DeepAugment+AugMix[18]	2020	No	53.6	24.2
Assemble-ResNet50[29]	2020	No	56.5	17.9
ANT (3×3)[37]	2020	Yes	63.0	23.9
BlurAfterConv [46]	2020	Yes	64.9	21.2
AugMix[20]	2020	Yes	65.3	22.5
MixUp+AA+AgMax	-	No	67.1	21.4
Stylized ImageNet[11]	2019	Yes	69.3	25.1
Patch Uniform[32]	2019	Yes	74.3	24.5
Baseline	-	N/A	76.7	23.8

Table 5. ImageNet-C robustness leaderboard with a ResNet50 backbone. Standalone indicates whether the method is a combination of techniques or a single method. Clean Error is the classification error on the uncorrupted validation set.

Data Augmentation	FGSM↑ $\epsilon = 0.1$	$\epsilon = 0.3$	$\epsilon = 0.5$
without AgMax			
Standard	24.9	13.4	8.0
CutOut	24.6	12.8	7.4
MixUp	31.8	21.0	15.1
CutMix	34.5	20.4	13.0
AA	29.3	20.4	15.3
CutOut+AA	28.9	20.2	14.4
MixUp+AA	35.0	26.4	20.7
CutMix+AA	37.0	27.2	20.3
with AgMax			
Standard	28.5	16.6	10.5
CutOut	26.2	14.3	9.0
MixUp	35.0	23.4	16.4
CutMix	37.5	22.3	13.5
AA	28.1	19.9	14.9
CutOut+AA	32.0	22.1	14.4
MixUp+AA	35.6	**28.1**	**23.5**
CutMix+AA	**37.7**	28.0	21.4

Table 6. Pre-trained (270 epochs) ResNet50 model Top-1% accuracy after FGSM attack with increasing strength, ϵ.

the corruption robustness of MixUp. The results show that while configurations with CutMix have low generalization errors, they perform poorly in terms of corruption robustness. To get an idea on how the best performing method *MixUp+AA+AgMax* fares in comparison with algorithms that are optimized for robustness, we borrowed the online leaderboard of Hendrycks *et al*. [19] as shown in Table 5. The performance of *MixUp+AA+AgMax* is competitive with the SOTA.

Table 6 shows that increasing the adversary strength ϵ on the validation set, *AgMax* generally improves the robustness of all pre-trained models by a wide margin. Similar to corruption robustness results, configurations that have good performance have MixUp in common. CutMix is also exhibiting robustness against FGSM. On the average, *MixUp+AA+AgMax* is also the best performing against adversarial attack.

4.5. Slower Training

Normalized to Standard model training time, *AgMax* incurs a performance penalty of about $1.7\times$ mainly due to the computation of the agreement between 2 positive samples.

Standard				CutOut				MixUp				CutMix			
MI	MSE	KL	CE	MI	MSE	KL	CE	MI	MSE	KL	CE	MI	MSE	KL	CE
77.4	**77.5**	77.3	77.4	79.2	**80.0**	78.9	79.2	79.5	78.7	**79.7**	79.3	80.0	**80.1**	80.0	80.0
AA				CutOut+AA				MixUp+AA				CutMix+AA			
MI	MSE	KL	CE	MI	MSE	KL	CE	MI	MSE	KL	CE	MI	MSE	KL	CE
79.2	**79.6**	79.0	79.0	81.1	**81.2**	81.1	80.9	**80.7**	80.1	80.5	80.5	**81.3**	80.8	81.2	80.9

Table 7. Top-1% accuracy on CIFAR100 of WideResNet40-2 trained for 200 epochs using different agreement functions: MI, MSE, KL-divergence and cross-entropy (CE).

MixUp			CutMix			MixUp+AA			CutMix+AA		
MI	MSE	KL	MI	MSE	KL	MI	MSE	KL	MI	MSE	KL
77.5	77.4	**77.7**	77.4	77.5	**77.6**	**77.1**	76.9	76.1	**77.0**	76.6	74.5

Table 8. Top-1% accuracy on ImageNet of ResNet50 trained for 90 epochs using different agreement functions: MI, MSE and KL-divergence.

Model	Stan	AM	CM	CM+AA	CM+AA+AM
RN101	78.1	79.2	79.8	80.7	**81.2**
RX200	68.8	**70.1**	65.6	63.5	65.5
RY400	74.3	**75.0**	71.4	70.6	71.8
EfNB0	75.3	**75.7**	72.3	71.7	71.9

Table 9. Top-1% accuracy on ImageNet using ResNet101 (RN101) for 270 epochs, RegNetX200MF (RX200), RegNetY400MF (RY400) and EfficientNet-B0 (EfNB0) all for 100 epochs as done in [35]. *Legend:* Standard (Stan), *AgMax-MI* (AM), AutoAugment (AA) and CutMix (CM). EfNB0-AM is *AgMax*-MSE.

Relative to Standard model, CutMix, MixUp, CutOut and AutoAugment have minimal impact on the training time.

4.6. Other Agreement Functions

Other agreement functions can be used in place of MI. Table 7 shows the comparison among MI, MSE, KL-divergence and CE on CIFAR100. In self-supervised learning, BYOL uses MSE while CE is the loss function in DINO. KL-divergence is also utilized since it is a good distance measure between two probability distributions. Table 7 shows that MSE is the best performing agreement function on Standard, CutOut, CutMix and AA. KL excels on MixUp. MI is the top performing method on MixUp+AA and CutMix+AA and has the highest overall performance. Table 8 shows that an ablation study on ResNet50, comparable results among MI, MSE and KL on MixUp and CutMix are observed. However, KL has a significantly lower performance on both MixUp+AA and CutMix+AA.

4.7. Other SOTA Models

Table 9 shows that other SOTA models benefit from *AgMax*. Large model such as ResNet101 gains +3.1% with CutMix+AA+*AgMax*. For RegNet [35] and EfficientNet [43], the use of *AgMax* improves the model accuracy. However, unlike in previous models, CutMix and AA perform poorly.

Dataset	Standard	CO+AA		MU+AA		CM+AA	
			+AgMax		*+AgMax*		*+AgMax*
	76.8	77.9	78.2	78.3	78.6	78.5	79.1
Faster R-CNN [36]							
PascalVOC	80.5	81.2	81.4	81.1	**81.5**	80.8	81.4
Mask R-CNN [16]							
COCO *box*	34.6	35.6	36.6	36.0	**37.0**	36.0	36.4
COCO *mask*	31.3	32.1	32.8	32.4	**32.9**	32.3	32.6

Table 10. mAP on object detection and segmentation tasks using ResNet50-FPN backbone trained for 270 epochs with Top-1% accuracy indicated for reference. CO is CutOut. MU is MixUp. CM is CutMix. AA is AutoAugment.

4.8. Object Detection and Instance Segmentation

Using the MMDetection framework [3], the performance of our pre-trained ResNet50 models on object detection and instance segmentation tasks on both PascalVOC [9, 8] and MS COCO datasets [31] can be evaluated. Table 10 shows that the backbone models pre-trained with *AgMax* consistently outperforms models without it. Similar to the results in robustness, models pre-trained with MixUp exhibit the best results in both detection and segmentation. Training Faster R-CNN [36] and Mask R-CNN [16] models uses the default MMDetection configurations with gradient clipping and $1\times$ schedule.

5. Conclusion

AgMax is a simple regularization technique that maximizes the agreement between the predictions of two positive samples. Empirical results demonstrated significant gains in performance on classification, object detection and segmentation.

6. Acknowledgement

This work was funded by the UP ECWRG 2019-2020, CHED-PCARI and ERDT-FRDG. Thanks to CNL and PCARI-PRIME: Roel Ocampo and Vladimir Zurbano, for server hosting.

References

[1] Mohamed Ishmael Belghazi, Aristide Baratin, Sai Rajeswar, Sherjil Ozair, Yoshua Bengio, Aaron Courville, and R Devon Hjelm. Mine: mutual information neural estimation. In *International Conference on Machine Learning*, 2018.

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *arXiv preprint arXiv:2104.14294*, 2021.

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[4] Pengguang Chen, Shu Liu, Hengshuang Zhao, and Jiaya Jia. Gridmask data augmentation. *arXiv preprint arXiv:2001.04086*, 2020.

[5] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Pecognition*, pages 113–123, 2019.

[6] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. *Advances in Neural Information Processing Systems*, 2020.

[7] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[8] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, Jan. 2015.

[9] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[10] Xavier Gastaldi. Shake-shake regularization. *International Conference on Learning Representations Workshops*, 2017.

[11] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019.

[12] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pages 10727–10737, 2018.

[13] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

[14] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc., 2020.

[15] Stephen José Hanson and Lorien Y Pratt. Comparing biases for minimal network construction with back-propagation. In *Advances in Neural Information Processing Systems*, pages 177–185, 1989.

[16] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[17] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[18] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[19] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

[20] Dan Hendrycks, Norman Mu, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2020.

[21] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. *International Conference on Learning Representations*, 2019.

[22] Daniel Ho, Eric Liang, Xi Chen, Ion Stoica, and Pieter Abbeel. Population based augmentation: Efficient learning of augmentation policy schedules. In *International Conference on Machine Learning*, pages 2731–2741, 2019.

[23] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European Conference on Computer Vision*, pages 646–661. Springer, 2016.

[24] Xu Ji, João F Henriques, and Andrea Vedaldi. Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9865–9874, 2019.

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[27] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Fractalnet: Ultra-deep neural networks without residuals. *International Conference on Learning Representations*, 2017.

[28] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[29] Jungkyu Lee, Taeryun Won, and Kiho Hong. Compounding the performance improvements of assembled techniques in a convolutional neural network. *arXiv preprint arXiv:2001.06268*, 2020.

[30] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In *Advances in Neural Information Processing Systems*, pages 6662–6672, 2019.

[31] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[32] Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. In *Workshop on Uncertainty and Robustness in Deep Learning at ICML*, 2019.

[33] XuanLong Nguyen, Martin J Wainwright, and Michael I Jordan. Estimating divergence functionals and the likelihood ratio by convex risk minimization. *IEEE Transactions on Information Theory*, 56(11):5847–5861, 2010.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[35] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10428–10436, 2020.

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[37] Evgenia Rusak, Lukas Schott, Roland S Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision*, 2020.

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations*, 2015.

[40] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[41] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2818–2826, 2016.

[42] Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Ricap: Random image cropping and patching data augmentation for deep cnns. In *Asian Conference on Machine Learning*, pages 786–798, 2018.

[43] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[44] Jonathan Tompson, Ross Goroshin, Arjun Jain, Yann LeCun, and Christoph Bregler. Efficient object localization using convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 648–656, 2015.

[45] Michael Tschannen, Josip Djolonga, Paul K Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. *International Conference on Learning Representations*, 2020.

[46] Cristina Vasconcelos, Hugo Larochelle, Vincent Dumoulin, Nicolas Le Roux, and Ross Goroshin. An effective anti-aliasing approach for residual networks. *arXiv preprint arXiv:2011.10675*, 2020.

[47] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[48] Yoshihiro Yamada, Masakazu Iwamura, Takuya Akiba, and Koichi Kise. Shakedrop regularization for deep residual learning. *IEEE Access*, 7:186126–186136, 2019.

[49] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.

[50] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *British Machine Vision Conference*, 2016.

[51] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *International Conference on Learning Representations*, 2018.

[52] Linjun Zhang, Zhun Deng, Kenji Kawaguchi, Amirata Ghorbani, and James Zou. How does mixup help with robustness and generalization? *arXiv preprint arXiv:2010.04819*, 2020.

[53] Xinyu Zhang, Qiang Wang, Jian Zhang, and Zhao Zhong. Adversarial autoaugment. In *International Conference on Learning Representations*, 2020.