A Self-Supervised Feature Map Augmentation (FMA) Loss and Combined Augmentations Finetuning to Efficiently Improve the Robustness of CNNs

Nikhil Kapoor  
Volkswagen AG  
Wolfsburg, Germany

Chun Yuan  
Volkswagen AG  
Wolfsburg, Germany

Jonas Löhdefink  
Technische Universität Braunschweig  
Braunschweig, Germany

Roland Zimmermann  
Volkswagen AG  
Wolfsburg, Germany

Serin Varghese  
Volkswagen AG  
Wolfsburg, Germany

Fabian Hüger  
Volkswagen AG  
Wolfsburg, Germany

Nic Schmidt  
Volkswagen AG  
Wolfsburg, Germany

Peter Schlicht  
Volkswagen AG  
Wolfsburg, Germany

Tim Fingscheidt  
Technische Universität Braunschweig  
Braunschweig, Germany

ABSTRACT
Deep neural networks are often not robust to semantically-irrelevant changes in the input. In this work we address the issue of robustness of state-of-the-art deep convolutional neural networks (CNNs) against commonly occurring distortions in the input such as photometric changes, or the addition of blur and noise. These changes in the input are often accounted for during training in the form of data augmentation. We have two major contributions: First, we propose a new regularization loss called feature-map augmentation (FMA) loss which can be used during finetuning to make a model robust to several distortions in the input. Second, we propose a new combined augmentations (CA) finetuning strategy, that results in a single model that is robust to several augmentation types at the same time in a data-efficient manner. We use the CA strategy to improve an existing state-of-the-art method called stability training (ST). Using CA, on an image classification task with distorted images, we achieve an accuracy improvement of on average 8.94% with FMA and 8.86% with ST absolute on CIFAR-10 and 8.04% with FMA and 8.27% with ST absolute on ImageNet, compared to 1.98% and 2.12%, respectively, with the well known data augmentation method, while keeping the clean baseline performance.

KEYWORDS
neural networks, robustness, data augmentation, safety, fine-tuning, convolutional neural networks

1 INTRODUCTION
Over the past few years deep neural networks (DNNs) have shown impressive performance on a variety of computer vision tasks such as image classification [15, 20, 22], object detection [13, 14, 23], semantic segmentation [6, 21, 29, 33], etc. However, recent works have demonstrated that these state-of-the-art networks are not robust to small changes in the input [2, 4, 5, 9, 10, 16, 24]. These small changes in the input, also called distortions, can be of various types, e.g., photometric changes (brightness, saturation, etc.) [31] or noise (Gaussian, salt and pepper (SAP) noise, etc.) [11]. In the real world, deviations from the training set distribution are to be expected. For instance, varying light conditions might affect the contrast and brightness of an image. Blurring of images might occur due to a shaky camera, bad weather conditions such as fog, rain, or simply incorrect camera settings (higher ISO, low shutter speed, etc.). Such changes in the input might lead to a wrong output of the model, as shown in Figure 1. In order to safely deploy neural networks in safety-critical situations such as autonomous cars or in bio-medical applications, it is vital to ensure high accuracy on the original task as well as high robustness of the output to small changes of the input.

Since there can be many different types of deviations arising due to many different reasons, it is quite challenging to train truly robust models that are invariant to all possible input changes. Traditionally, data augmentation is performed in order to overcome
this challenge [30]. This means that augmented images are added into the training set and the model is finetuned on this extended training set while keeping the original loss. However, there seems to be no clear understanding on how effective this approach truly is. It also suffers from the obvious downside of increased computational expense as the size of the dataset increases proportionally to the number of augmentations. Azulay et al. [2] also show that data augmentation merely leads to an increased invariance to augmentation types only for images that look very similar to typical training set images. This leads us to the question: “Can we improve on data augmentation and come up with a better way to ensure robustness of convolutional neural networks to commonly occurring image corruption types without the downside of increased computational expense?”

We aim to answer this question in the scope of this paper.

For this, we propose a novel feature map augmentation (FMA) loss as well as a new combined augmentation (CA) training strategy that aims at increasing robustness of convolutional neural networks to a pre-defined set of commonly occurring augmentation types in a data-efficient manner. Our method outperforms data augmentation by a large margin over a range of augmentation types on two different classification datasets, namely ImageNet [20] and CIFAR-10 [1], while maintaining its original task performance. We test our approach over five different augmentation types as shown in Table 1. In summary, our contributions are as follows:

1. We propose an additional feature map augmentation (FMA) loss term that aims at making any given pre-trained convolutional neural network (CNN) model robust to a predefined set of input distortions using only a few subsequent epochs of finetuning.
2. For training a model with multiple augmentation types at the same time, we propose a new data-efficient combined augmentation (CA) training strategy and use this to additionally improve on an existing state-of-the-art method for robust training.
3. Finally, we demonstrate that when compared to data augmentation, our finetuned model is significantly more robust to multiple augmentation types at the same time and also keeps its original classification accuracy.

2 RELATED WORK

This section highlights existing work on data augmentation, other robustness enhancement techniques in general, and stability training in particular.

2.1 Data augmentation

Data augmentation is common practice in neural network training where training samples are augmented with different augmentation types and the network is trained on this extended data set [18, 28]. Although it helps increase generalization, the computational complexity also increases. For the sake of this paper, we term this method as augmentation training (AT). Several approaches have been proposed to selecting clever augmentation policies such as Autoaugment [8], AugMix [17], Randaugment [7] etc., however most of these approaches tend to add additional computational overhead of searching for an effective augmentation strategy. More so, while these approaches tend to increase generalization, they are not particularly efficient in improving robustness to a held out set of augmentations. In contrast, we aim at improving robustness of an already well trained model, given a set of pre-defined relevant augmentations.

2.2 Robustness enhancements

In order to improve classifier stability, Vasilijevi et al. [27] finetuned on blurred images. To generalize to other blurs, they found that it is not enough to finetune on one type of blur to generalize to other blur types. On the other hand, Rosza et al. [25] proposed a simple training technique called batch-adjusted network gradients (BANG) that does not need any additional training data to improve robustness. They propose a slight variation of batch normalization by balancing weight updates which inherently increases robustness in general by smoothing the decision boundaries. However, their approach is not self-adaptive to other models and tasks. Geihros et al. [12] proposed Stylized ImageNet, where clean images were converted to different styles/textures, such as canvas paints and the model was trained on these stylized images in addition to clean images. In order to evaluate corruption robustness, Hendrycks et al. [16] proposed a public benchmark of 15 corruption types at 5 different severities on CIFAR-10 and ImageNet dataset, however, this dataset is only meant to be tested for methods that do not explicitly train on the same augmentations. For this reason, we do not use their benchmark for evaluation of our results.

2.3 Stability training (ST)

Zheng et al. [32] introduced an additional regularization loss which penalizes the prediction difference of the softmax output of clean and perturbed images. Additionally, they proposed to train only on images augmented with Gaussian Noise (GN) as a means of improving robustness in a general manner to many augmentation types. However, on re-implementation of their method, we could not confirm this to be true: We noticed that training with GN only helped improve robustness to GN and other noise types such as

| Augmentation       | Parameters $\phi_n$ |
|-------------------|---------------------|
| Brightness+       | $\Delta = 0.39$     |
| Brightness−       | $\Delta = -0.36$    |
| Saturation+        | $\alpha = 6.0$      |
| Saturation−        | $\alpha = 0.0$      |
| Gaussian noise    | $\mu = 0.0, \sigma = 0.075$ |
| Gaussian blurring | $s = 3.0, \mu = 0.0, \sigma = 0.675$ |
| Additive SAP noise| $p = 0.025, q = 0.5, \rho = 0.5$ |

| Augmentation       | Parameters $\phi_n$ |
|-------------------|---------------------|
| Brightness+       | $\Delta = 0.43$     |
| Brightness−       | $\Delta = -0.32$    |
| Saturation+        | $\alpha = 4.0$      |
| Saturation−        | $\alpha = 0.2$      |
| Gaussian noise    | $\mu = 0.0, \sigma = 0.08$ |
| Gaussian blurring | $s = 3.0, \mu = 0.0, \sigma = 1.175$ |
| Additive SAP noise| $p = 0.01, q = 0.7, \rho = 0.7$ |
Figure 2: Finetuning with new total loss, including feature map augmentation (FMA) loss. A CNN model pre-trained using the original task loss \( J_0 \) (5) can be finetuned using the loss (4) including the new regularization loss called feature map augmentation (FMA) loss \( J_{\text{FMA}} \) (6). The FMA loss is computed by a self-supervised regularization of differences of feature activation maps of all layers to clean and augmented image pairs. The hyper-parameter \( \gamma \) controls the trade-off between clean and augmented accuracy.

salt and pepper, with a corresponding robustness decrease in other augmentation types. Hence, to improve further on ST [32] and conventional data augmentation (AT) [30], we propose a new combined augmentation (CA) training strategy which will be discussed later.

3 FEATURE MAP AUGMENTATION (FMA) LOSS & TRAINING STRATEGY

In this section, we first describe the intuition behind the idea of the feature map augmentation (FMA) loss. We then present details of our method and show how this can be used to stabilize feature embeddings.

3.1 Intuition

In an ideal world, a CNN should respond similarly to a clean and an augmented image as long as the semantic content of the images remains the same, i.e., the network is expected to be augmentation-invariant. Therefore, we would hope that the feature activation map of individual layers should also remain the same, given a clean and its corresponding augmented input. If not, we can assume that the filters corresponding to the deviations in feature maps are sensitive to augmentations. In consequence, we will propose a new feature map augmentation (FMA) loss which regularizes the normalized difference in feature maps between a clean and an augmented image. We expect that this would increase the model robustness as the robustness objective is now made explicit and the model is optimized towards achieving this goal.

3.2 Robustness Objective

In this section we define our robustness objective that we optimize for. First, however, we define some mathematical notation.

Let \( \mathcal{A} = \{A_1, \ldots, A_n, \ldots, A_N\} \) be a set of \( N \) augmentation types, as also shown in Table 1. We define \( x \in \mathbb{R}^{H \times W \times C} \) as an image of dataset \( X \) with \( \mathbb{G} = \{0, 1\} \) being the set of gray values, image height \( H \), image width \( W \), and number of color channels \( C \). The image \( x \) is fed into a neural network \( \tilde{g}(x; \theta) \) having the network parameters \( \theta \). The neural network \( \tilde{g}(x; \theta) \) consists of several layers \( t \in \mathcal{L} = \{1, 2, \ldots, L\} \), each having an output feature map \( f_t(x) \in \mathbb{R}^{n \times H_t \times W_t \times C_t} \) (assuming a ReLU activation function) with the height \( H_t \), width \( W_t \) and number of feature maps \( C_t \), and \( \mathbb{R}^{\geq 0} = \{x \in \mathbb{R} \mid x \geq 0\} \).

Given this notation, we can write the overall computation of the neural network \( \tilde{g}(x; \theta) \) as follows:

\[
\tilde{y} = \tilde{g}(x; \theta) = a(f_t(f_{t-1}(\ldots(f_2(f_1(x)))))))
\] (1)

where \( f_t \) denotes the feature map tensor of layer \( t \), where \( t \in \mathcal{L} \), and \( a(\cdot) \) denoting the output layer performing a scalar prediction. For each augmentation type \( A_n \), we can compute the corresponding augmented image as follows:

\[
\tilde{x} = \delta_n(x) \in \mathbb{G}^{H \times W \times C},
\] (2)

where \( \delta_n(\cdot) \) is the augmentation function with specific parameters (see Table 1) \( \phi_n \in \phi = \{\phi_1, \ldots, \phi_N\} \), corresponding to the augmentation type \( A_n \in \mathcal{A} \). Irrespective of the applied augmentation type, function \( \delta_n(\cdot) \) always performs clipping of each pixel to enforce \( \delta_n(\cdot) \in \mathbb{G}^{H \times W \times C} \). This allows handling zero-mean noise as well. In order to make our model robust to all the augmentations, we want to ensure that the feature maps are similar for both clean as well as distorted images for all the augmentations. The robustness objective is therefore,

\[
\forall(x, \tilde{x}) : f_t(x) \approx f_t(\tilde{x}), t \in \mathcal{L}
\] (3)

with \( \tilde{x} \) as in (2). Given an existing training objective \( J_0 \) on the original task (e.g., classification), an input image \( x \) and a perturbed copy \( \tilde{x} \), we can implement the new total loss \( J \) for finetuning with the robustness objective (3) as:

\[
J(x, \tilde{x}; \theta) = J_0(x; \theta) + \gamma J_{\text{FMA}}(x, \tilde{x}; \theta),
\] (4)

where \( \gamma \) controls the strength of the regularization term \( J_{\text{FMA}} \). In terms of classification, the \( J_0 \) term can be a standard cross-entropy loss

\[
J_0(x; \theta) = -\sum_{j \in \mathcal{J}} \tilde{y}_j \log P(y_j|x; \theta),
\] (5)

where the index \( j \in \mathcal{J} \) runs over the number of classes and \( \tilde{y}_j \in \{0, 1\} \) is a binary indicator being 1 if the predicted class label \( j \) is the correct classification and 0 otherwise. The new loss term in (4) can then be defined as

\[
J_{\text{FMA}}(x, \tilde{x}; \theta) = \frac{1}{|L|} \sum_{t \in \mathcal{L}} |f_t(x) - f_t(\tilde{x})|^2,
\] (6)
We now demonstrate our two-stage training approach using the ImageNet dataset. The hyperparameter $\psi$ leads to a drop of absolute performance on the validation set, except Saturation$^+/−$ and Combined$^+/−$. Each augmentation leads to a drop of $10\%$ absolute performance on the validation set, except Saturation$^+/−$ and Combined$^+/−$. Top row: CIFAR-10 clean and augmented example image. Bottom row: ImageNet clean and augmented example image.

Figure 3: Qualitative visualization of augmentation types as introduced in Table 1. $'+'$ and $'−'$ here denote increase and decrease, respectively. Combined$^+$ means all augmentations applied at the same time except Brightness$^−$ and Saturation$^−$. Combined$^−$ means all augmentations applied at the same time except Brightness$^+$ and Saturation$^+$. Each augmentation leads to a drop of $10\%$ absolute performance on the validation set, except Saturation$^+/−$ and Combined$^+/−$. Our experiments are performed for the image classification task on two well-known datasets, namely CIFAR-10 [1] and a subset of ImageNet [20]. The CIFAR-10 dataset consists of 50,000 training images and 10,000 test images being $32 \times 32$ color images sorted in 10 classes. The ImageNet dataset (ILSVRC 2012) on the other hand consists of a total of 1.2 million training images and 50,000 validation and 150,000 test samples being $224 \times 224$ color images sorted in 1000 classes. For the sake of computational ease, we consider a subset of 200 randomly chosen classes from the ImageNet dataset instead. This reduces the number of images to 240,000 training images and 10,000 test images, thereby accelerating our experiments significantly. For both of these datasets, we consider a standard VGG-16 [26] model pre-trained on ImageNet weights (downloaded from the official tf.slim repository$^1$). The model is adapted to both the datasets in terms of its input and output dimensions, and is trained using stochastic gradient descent with momentum optimizer for an additional 40 epochs with varying learning rates $\eta_0 = (10^{-2}, 10^{-4}, 10^{-6})$ for $(20, 10, 10)$ epochs, respectively. With this configuration, we achieve a baseline accuracy of $89.82\%$ and $79.77\%$ on the validation set of the CIFAR-10 and the ImageNet dataset, respectively. We use these baseline models for all our experiments.

4.1 Network and Dataset

In this section, we explain the augmentation types (see Tables 1) considered in this work and a few implementation details. Remember that pixel-wise clipping is finally performed in each of the augmentation functions $\delta_n(\cdot)$ in (2). We also introduce combinations of these augmentations that are used for later experiments. Combinations of augmentations always follow (7) with additional final pixel-wise clipping.

4.2 Augmentation Types and Parameters

In this section, we first present implementation relevant details such as augmentation types, the network and datasets used with corresponding hyperparameters and then introduce our novel combined augmentation (CA) training strategy.

Our experiments are performed for the image classification task on two well-known datasets, namely CIFAR-10 [1] and a subset of ImageNet [20]. The CIFAR-10 dataset consists of 50,000 training images and 10,000 test samples being $32 \times 32$ color images sorted in 10 classes. The ImageNet dataset (ILSVRC 2012) on the other hand consists of a total of 1.2 million training images and 50,000 validation and 150,000 test samples being $224 \times 224$ color images sorted in 1000 classes. For the sake of computational ease, we consider a subset of 200 randomly chosen classes from the ImageNet dataset instead. This reduces the number of images to 240,000 training images and 10,000 test images, thereby accelerating our experiments significantly. For both of these datasets, we consider a standard VGG-16 [26] model pre-trained on ImageNet weights (downloaded from the official tf.slim repository$^1$). The model is adapted to both the datasets in terms of its input and output dimensions, and is trained using stochastic gradient descent with momentum optimizer for an additional 40 epochs with varying learning rates $\eta_0 = (10^{-2}, 10^{-4}, 10^{-6})$ for $(20, 10, 10)$ epochs, respectively. With this configuration, we achieve a baseline accuracy of $89.82\%$ and $79.77\%$ on the validation set of the CIFAR-10 and the ImageNet dataset, respectively. We use these baseline models for all our experiments.

4 IMPLEMENTATION DETAILS

In this section, we first present implementation relevant details such as augmentation types, the network and datasets used with corresponding hyperparameters and then introduce our novel combined augmentation (CA) training strategy.

$^1$https://github.com/tensorflow/models/tree/master/research/slim#Pretrained
Table 2: Accuracy values for individual augmentations both in training (IA strategy) and test (Section 5.1) on the CIFAR-10 validation set. For each augmentation type, a baseline model is finetuned with three different methods, i.e., augmentation training (AT), stability training (ST), or by our new FMA-based total loss (6). The evaluation is performed on both clean and augmented validation sets for the same augmentation type that it was finetuned for. ACC1 and ACC2 refer to accuracy of a model evaluated on the clean validation set before and after robust training, respectively. Similarly, ACC1 and ACC2 refer to the accuracy of a model evaluated on the augmented validation set before and after robust training, respectively.

| Accuracy | Training Method | Augmentation Type $A_n$ | Average Improvement |
|----------|-----------------|-------------------------|---------------------|
| Clean    |                 | B+ B− GB GN SAP S+ S−   | ACC$_2$ − ACC$_1$   |
| ACC$_1$  |                 | 89.82% 89.82% 89.82%    |                    |
| AT       |                 | 90.35% 90.26% 90.00%    | 0.13%              |
| ST       |                 | 90.58% 90.32% 90.56%    | 0.65%              |
| FMA      |                 | 90.20% 90.33% 90.23%    | −0.03%             |
| Augmented|                 |                        | ACC$_2$ − ACC$_1$   |
| ACC$_1$  |                 | 80.13% 80.13% 81.06%    |                    |
| AT       |                 | 87.80% 86.04% 89.17%    | 6.44%               |
| ST       |                 | 88.50% 86.97% 89.39%    | 7.11%               |
| FMA      |                 | 88.69% 87.47% 89.32%    | 7.32%               |

Δ to each of the RGB channels. Saturation ($S$) changes can be implemented by multiplying the image’s saturation in the hue, saturation and luminance (HSL) representation by a factor $\alpha$.

4.2.2 Noise and blurring augmentation types. Gaussian noise (GN) can occur due to sensor noise by poor illumination and/or high temperature, etc. Additive salt-and-pepper noise (SAP) can occur due to bit errors in image transmission [16]. Gaussian blurring (GB) can occur due to out-of-focus images or incorrect camera configuration. For implementation, the Gaussian noise is zero-mean with standard deviation $\sigma$. In contrast to this, additive salt-and-pepper noise adds a binary noise with strength $\pm p$ to the image, where the probability of the noise is given by $p$ and the salt-to-pepper ratio by $q$. For the Gaussian blur, a zero-mean Gaussian kernel of size $s \times s$ with standard deviation $\sigma$ is convoluted over the image $x$.

4.3 Training Strategies

We employ two training strategies for our experiments that are discussed next.

4.3.1 Individual augmentation (IA) training strategy. Individual augmentation refers to the strategy of augmenting each image with only a single augmentation type one at a time. Hence, multiple augmentation types would mean multiple copies of the original image, each augmented with a different augmentation type. This increases the size of the dataset proportionally to the number of augmentation types considered. The strength of the augmentations is chosen such that the validation accuracy drops by roughly 10%.

4.3.2 Combined augmentation (CA) training strategy. As the number of augmentation types increases, so does the dataset size. Training on this extended data set can involve high computational cost. In order to be data-efficient, we propose an alternate approach called CA training strategy. In contrast to IA, this means augmenting the same image with multiple augmentation types all at once. Hence, as the number of augmentation types increases, the dataset size does not explode, and hence being more efficient. However, a naive combination of all augmentations at once can be counter-effective as some augmentation types (such as increase/decrease of brightness) can cancel each other out. We term such augmentations as mutually inverse augmentations. In order to counter the inverse effect, we propose to create two sets of such augmentations following (7). We describe this in more detail now. However, before doing so, we define a few notations.

Consider the set of all individual augmentations denoted by $A = \{B+, B−, S+, S−, GN, SAP, GB\}$ where ‘+’ and ‘−’ indicate an increase and decrease, respectively. Assuming the set of non-inverse augmentations defined as $A' = \{GN, SAP, GB\}$, we can reasonably group the inverse augmentations as “Combined+”, containing $\{B+, S+\} \cup A'$, and “Combined−”, containing $\{B−, S−\} \cup A'$. Given such a grouping, during training, we simply alternate between Combined+ and Combined− every epoch, such that both the inverse augmentations do not cancel each other out in this way. One could also alternate images every batch, however from our experiments, this led to sub-optimal results. Lastly, if we increase the number of epochs between the altering augmentation sets, this would lead to catastrophic forgetting [19].

5 EXPERIMENTS AND RESULTS

The experiments in this section are split into two parts. The first part deals with testing our method against individual augmentations $\hat{x}$ as defined in (2) and comparing our method with existing state-of-the-art robustness methods. The model is trained on an augmentation type, say $A_1$, and then tested on the same augmentation type during evaluation. The second part deals with evaluating the effectiveness of our combined augmentation (CA) training strategy on different methods. The aim is to investigate if training using CA helps improve robustness on individual augmentations separately.

5.1 Training With Individual Augmentations

As a first experiment, we investigate the increase in robustness of our model finetuned to seven individual augmentations (as shown in Table 1) separately. We test our method with existing state-of-the-art robustness methods, namely augmentation training (AT) and stability training (ST) and show results in Table 2. For the
Table 3: Accuracy values for models finetuned with combined augmentations (AT/CA, ST/CA and FMA/CA) by using multiple augmentations simultaneously. The baseline model, as well as the three models after finetuning, are then evaluated with individual augmentations separately, as well as with clean images on the validation set. The strength of augmentations used in the combined training approach is such that most augmentation types individually lead to a performance drop of roughly 10%. Therefore, when applied together, the combined accuracy drop is much more severe than 10%.

|                | CIFAR-10 validation set | ImageNet validation set |
|----------------|-------------------------|-------------------------|
|                | Baseline | AT/CA | ST/CA | FMA/CA | Baseline | AT/CA | ST/CA | FMA/CA |
| Clean          | 89.82%   | 88.14%| 89.91%| 88.97% | 79.77%   | 78.14%| 81.01%| 81.19% |
| Brightness+ (B+) | 80.13%   | 75.92%| 82.32%| 82.23% | 69.72%   | 64.89%| 73.66%| 70.00% |
| Brightness− (B−) | 80.13%   | 74.40%| 81.14%| 81.84% | 68.91%   | 65.12%| 70.18%| 70.77% |
| Gaussian blur (GB) | 81.06%   | 80.31%| 83.71%| 85.02% | 70.59%   | 68.24%| 70.72%| 71.75% |
| Gaussian noise (GN) | 81.17%   | 87.34%| 86.53%| 86.94% | 70.81%   | 75.34%| 74.03%| 75.44% |
| Additive SAP noise (SAP) | 80.73%   | 86.94%| 85.72%| 86.58% | 72.32%   | 78.02%| 75.90%| 76.99% |
| Saturation+ (S+) | 80.47%   | 73.71%| 84.38%| 83.74% | 67.30%   | 64.25%| 68.63%| 66.98% |
| Saturation− (S−) | 84.03%   | 82.03%| 84.44%| 83.70% | 64.15%   | 61.53%| 66.87%| 69.61% |
| Combined+      | 44.22%   | 54.62%| 74.08%| 73.50% | 18.46%   | 33.83%| 58.06%| 54.93% |
| Combined−      | 29.87%   | 48.10%| 67.98%| 68.59% | 13.47%   | 27.43%| 39.15%| 38.27% |
| Average improvement | − 1.98%| 8.86% | 8.94% | − 2.12%| 8.27% | 8.04% |
on both datasets, AT performs worse than the baseline model on all augmentation types, except on GN and additive SAP noise. The augmented accuracy gain with AT on GN, and SAP noise is mainly due to the fact that in our CA training strategy, GN and SAP noise are seen in every epoch (see Section 4.3). As expected, the gain in augmented performance on mutually inverse augmentations such as B+/- and S+/- is relatively lower for all the three methods. On the CIFAR-10 dataset, FMA performs better than ST with an average improvement of 8.94%, compared to 8.85%. On the ImageNet dataset, however, ST performs better than FMA with an average improvement of 8.27% compared to 8.04%, respectively. On the other hand, AT only improves by an average of 1.98% and 2.12% on the CIFAR-10 and ImageNet dataset, respectively. These results clearly show the effectiveness of our CA training strategy in terms of increasing robustness to multiple augmentations simultaneously while being data-efficient.

Lastly, we observe the improvement of individual augmentation performance as training progresses for our FMA loss model in Figure 4. We report highest improvement in the set of non-inverse augmentations \( \mathcal{A} \) such as GN, additive SAP noise and GB and relatively lower improvement in the set of inverse augmentations. This is primarily because the non-inverse augmentations are seen more often due to the training strategy itself when compared to the inverse augmentations such as B+/- and S+/- . The performance on clean data is also more or less recovered.

6 CONCLUSION

In this work, we proposed a new feature map augmentation (FMA) loss which can be used to efficiently stabilize a CNN to a variety of commonly known input distortions. We also introduced a new combined augmentation (CA) training strategy, which can be used to gain robustness to multiple augmentations at once in a data-efficient manner. Using CA, we further improved an existing state-of-the-art method called stability training (ST) [32]. In the end, for both CIFAR-10 and ImageNet datasets, we attained a single model each, which have an average augmented accuracy improvement of 8.94% and 8.04% absolute, respectively, while retaining original task performance. In comparison, conventional data augmentation only achieves 1.98% and 2.12%, respectively. These results indicate that clever combinations of data augmentations, together with additional robustness-focused loss functions, can help improve robustness in a data-efficient manner towards a held-out set of relevant corruptions. This is significantly better than conventional data augmentation.

As a scope of future work, it would be interesting to find underlying similarities between several augmentation types and study their generalization abilities. Based on these clever augmentation sub-sets, FMA loss can be applied on top to attain robust models that also generalize well to unseen augmentations.

ACKNOWLEDGMENTS

The research leading to these results is funded by the German Federal Ministry for Economic Affairs and Energy within the project “KI Absicherung – Safe AI for Automated Driving”. The authors would like to thank the consortium for the successful cooperation.

REFERENCES

[1] A. Krizhevsky. 2009. Learning Multiple Layers of Features from Tiny Images. https://www.cs.toronto.edu/~kriz/cifar.html. Accessed: 2019-10-10.
[2] Aharon Azulay and Yair Weiss. 2018. Why Do Deep Convolutional Networks Generalize so Poorly to Small Image Transformations? arXiv:1805.12177 (May 2018).
[3] James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research (JMLR) 13, 1 (March 2012), 281–305.
[4] Charlotte Bunne, Lukas Rahmann, and Thomas Wolf. 2018. Studying Invariances of Trained Convolutional Neural Networks. arXiv:1803.05963 (March 2018).
[5] Andreas Bär, Fabian Hüger, Peter Schlicht, and Tim Fingscheidt. 2019. On the Robustness of Teacher-Student Frameworks for Semantic Segmentation. In Proc. of CVPR - Workshops. Long Beach, CA, USA, 1–9.
[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv:1706.05587 (June 2017).
[7] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. 2020. Randaugment: Practical Automated Data Augmentation with a Reduced Search Space. In Proc. of CVPR. Seattle, WA, USA, 702–703.
[8] Ekin D. Cubuk, Barret Zoph, Dandelion Man, Vijay Vasudevan, and Quoc V. Le. 2019. AutoAugment: Learning Augmentation Strategies from Data. In Proc. of CVPR. Long Beach, CA, USA, 113–123.
