This WACV paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Benchmark Generation Framework with Customizable Distortions for Image Classifier Robustness

Soumyendu Sarkar[†*] Ashwin Ramesh Babu[†] Sajad Mousavi[†] Zachariah Carmichael[†]
Vineet Gundecha Sahand Ghorbanpour Ricardo Luna Gutierrez Antonio Guillen
Avisek Naug
Hewlett Packard Enterprise, USA
{soumyendu.sarkar, ashwin.ramesh-babu, sajad.mousavi, zachariah.carmichael}@hpe.com
{vineet.gundecha, sahand.ghorbanpour, rluna, antonio.guillen, avisek.naug}@hpe.com

Abstract

We present a novel framework for generating adversarial benchmarks to evaluate the robustness of image classification models. Our framework allows users to customize the types of distortions to be optimally applied to images, which helps address the specific distortions relevant to their deployment. The benchmark can generate datasets at various distortion levels to assess the robustness of different image classifiers. Our results show that the adversarial samples generated by our framework with any of the image classification models, such as ResNet-50, Inception-V3 and VGG-16, are effective and transferable to other models causing them to fail. These failures happen even when these models are adversarially retrained using state-of-the-art techniques, demonstrating the generalizability of our adversarial samples. We achieve competitive performance in terms of net L_2 distortion compared to state-of-the-art benchmark techniques on CIFAR-10 and ImageNet; however, we demonstrate that our framework achieves such results with simple distortions like Gaussian noise without introducing unnatural artifacts or color bleeds. This is made possible by a model-based reinforcement learning (RL) agent and a technique that reduces a deep tree search of the image for model sensitivity to perturbations, to a one-level analysis and action. The flexibility of choosing distortions and setting classification probability thresholds for multiple classes makes our framework suitable for algorithmic audits.

1. Introduction

Neural networks' susceptibility to adversarial perturbations has raised concerns about their reliability. Adversarial perturbations are slight alterations to input data that can cause neural networks to make confident yet incorrect predictions. Despite efforts to understand and counter adversarial perturbations, existing defense strategies have shown limited improvements in robust accuracy. This emphasizes the need for alternative approaches to evaluate and enhance neural network robustness. Recent research suggests that generating additional subsets from the main dataset through perturbations/augmentations can improve robustness in fully-supervised and semi-supervised settings [14]. To utilize the original training set more effectively, modifications are introduced. One popular recent approach, proposed by Hendrycks and Dietterich (2018), aims to evaluate model robustness and ultimately enhance it [14].

We propose a machine learning-driven adversarial data generator that introduces natural distortions to create an adversarial subset from an original dataset. Our approach formulates the generation of adversarial samples as a Markov Decision Process (MDP). By dividing the input sample into patches, we aim to identify and add distortions to the most vulnerable areas, leading to misclassification. Our generator utilizes an addition and removal mechanism, mimicking a deep tree search to find vulnerabilities and add noise in the right locations. Additionally, our method allows users to incorporate custom datasets and distortion types for generating adversarial samples.

As part of our work, we provide adversarial subsets derived from CIFAR-10 and ImageNet datasets. We evaluated the performance of adversarially trained models using state-of-the-art techniques from the literature on our dataset. The performance of these models on our dataset is noticeably lower than on the clean dataset and a competitor's benchmark [14]. We achieved an average L_2 value of 2.48 (evaluated over 1,000 ImageNet samples) and a maximum of 4.74. Our benchmark will assist future initiatives in building robust architectures, which is crucial considering

*Corresponding author. †Equal contribution.

the increasing concerns and requirements for robust deep-learning models.

The main contributions of this paper are as follows:

- We propose a framework to generate adversarial benchmarks with a custom mix of distortions for evaluating the robustness of image classification models against both true negatives and false positives.
- We enable robustness audits for distortions characteristic of use cases at deployment for multiple distortion thresholds.
- We achieve competitive performance with the state-of-the-art on multiple metrics of minimum distortions needed for misclassification.
- We are competitive with the state-of-the-art on improving robustness with adversarial training.

2. Related Works

2.1. Data augmentation and adversarial samples for improving robustness

Several data augmentation techniques have been proposed to enhance the robustness of deep learning models. Cutout [8] masks out regions of input images which forces models to rely on alternative informative features. Mixup [33] generates virtual training samples by interpolating between pairs of images and labels, reducing overfitting and increasing robustness. Manifold Mixup [29] extends this idea by interpolating between feature representations. CutMix [32] combines Cutout and Mixup by replacing masked regions with patches from other images. AugMix [15] applies diverse augmentations to images, encouraging models to learn from a wide range of variations. Randaugment [7] applies random sequences of augmentation policies. RandConv [31] applies random convolutions as data augmentation. ALT [12] uses adversarially learned transformations to obtain both objectives of diversity and hardness at the same time. AutoAugment [6] and other recent works [22–26] uses Reinforcement Learning (RL) to discover optimal data augmentation policies. These techniques manipulate training data through various transformations, improving the models' robustness and generalization to adversarial perturbations.

2.2. Adversarial training for improved robustness

Recent research has explored various approaches to improve the robustness and out-of-distribution (OOD) performance of deep networks. Diffenderfer et al. [9] focused on compressing deep networks to enhance OOD robustness, demonstrating improved performance in handling OOD samples through network compression techniques. Kireev et al. [16] investigated the effectiveness of adversarial training against common corruption, identifying the strengths and limitations of this approach. They explored the performance of adversarially trained models and suggested areas for improvement. Modas et al. [19] proposed PRIME, a framework that leverages primitive transformations during training to enhance robustness against common corruptions, achieving significant improvements in model performance on corrupted inputs. Wang et al. [30] introduced better diffusion models in adversarial training to enhance its effectiveness against adversarial attacks. Tian et al. [28] conducted a comprehensive analysis of the robustness of Vision Transformers (ViTs) towards common corruptions. Geirhos et al. [11] presented a study on the bias toward texture in ImageNet-trained Convolutional Neural Networks (CNNs), showing their reliance on texture rather than shape cues. Erichson et al. [10] developed NoisyMix, a framework that combines data augmentations, stability training, and noise injections to improve the robustness of deep neural networks.

2.3. Benchmark to evaluate robustness

Data augmentation techniques and benchmark datasets play a crucial role in evaluating and enhancing the robustness of image classification models. Hendrycks and Dietterich [14] introduced multiple datasets based on ImageNet and used them as benchmarks for evaluating the robustness of models to input corruptions. **ImageNet-C** contains common visual corruptions applied to the ImageNet dataset and allows researchers to assess model performance under various types of visual distortions. **ImageNet-A** focuses on evaluating robustness to common image corruptions by providing a standardized evaluation environment. **ImageNet-P**, on the other hand, assesses the vulnerability of models to subtle perturbations by introducing imperceptible changes to deceive the models while maintaining visual similarity. The **Adversarial Robustness 101 (AR101)** benchmark [5] provides a comprehensive evaluation of model robustness against different attack types using the CIFAR-10 and CIFAR-100 datasets. PACS, Office-Home, MNIST-C, and WILDS benchmark datasets [1, 18, 20, 21] are designed to evaluate the domain adaptation and out-of-distribution robustness of the models. Lastly, the **Robustness via Dataset Manipulation (RoD)** [27] benchmark focuses on evaluating adversarial robustness against physical-world attacks by including real-world images with physical modifications. These benchmarks enable researchers to compare the performance of models and defense techniques in challenging scenarios.

3. Design of the Benchmark Generator

The evaluation for a machine learning model can be represented as $y = \text{argmax} f(x; \theta)$, where x denotes the in-

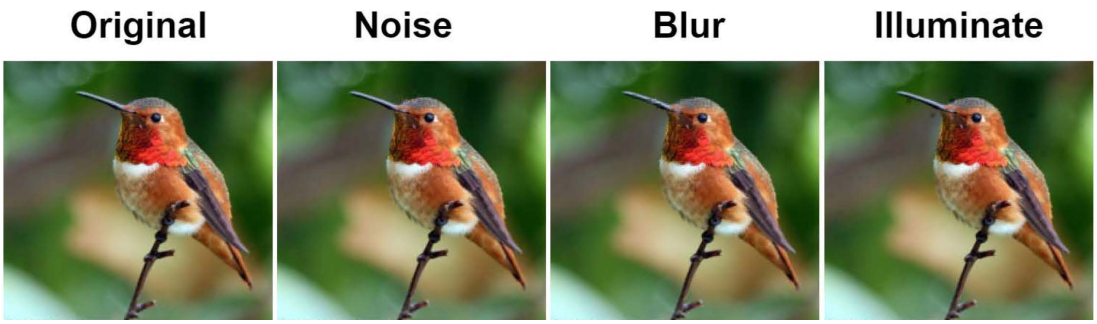


Figure 1. Adversarial samples with multiple distortion types (original picture from ImageNet)

put image, y represents the prediction, θ represents the model parameters, and the function f represents the machine learning model's output,

3.1. Markov Decision Process (MDP) formulation

3.1.1 MDP for un-targeted attack

An un-targeted black-box adversarial sample generator, used for true negative evaluation, without access to the θ, generates a perturbation δ such that $y_{\text{true}} \neq f(x+\delta;\theta)$. L_p norms specify the distance between the original and the adversarial sample, $D(x, x+\delta)$. Our objective is to cause misclassification while keeping D to a minimum.

State S_t contains a number of lists related to the classification probability and sensitivity of the image regions. Action A_t represent the perturbation to obtain the adversarial sample defined as:

$$A_t : x \rightarrow x + \delta_t, \tag{1}$$

where δ_t defines the perturbation at time step t, or more specifically which patches of the original sample x are going to be distorted. We define a probability dilution (PD) metric, which measures the extent to which the classification probability shifts from the ground truth to the other classes. The difference between the PD of the altered and the original image as a result of an action at each step (ΔPD), is a measure of the effectiveness of the action. Moreover, the change in L_2 distance (ΔL_2) as a measure of the distortion added is the cost for action. The reward is defined by the normalized PD as represented in equation 2.

$$R_t = \Delta\text{PD}_{\text{norm}} = \Delta\text{PD}/\Delta L_2 \tag{2}$$

The change in the distribution of the probabilities across classes is updated in the state vector at every step such that the RL agent can choose the optimum action at every step, maintaining the L_p and the number of steps (queries).

3.1.2 MDP for targeted attack

A targeted black-box attack, used for false positive evaluation, without access to the θ generates a perturbation δ such that $y_{\text{target}} = f(x+\delta;\theta)$ s.t. $y_{\text{target}} \neq y_{\text{true}}$. L_p norms specify the distance between the original and the adversarial sample, $D(x, x+\delta)$. Our objective is to cause misclassification while keeping D to a minimum. The action A_t will be defined as in equation 1.

We define a probability enhancement (PE) metric, which measures the extent to which the classification probability of the non-ground truth target class goes up. The difference between the PE of the altered image and the original image as a result of an action at each step (ΔPE), is a measure of the effectiveness of the action. Moreover, the change in L_2 distance (ΔL_2) as a measure of the distortion added is the cost for action. The reward is defined by the normalized PE as represented in equation 3.

$$R_t = \Delta\text{PE}_{\text{norm}} = \Delta\text{PE}/\Delta L_2 \tag{3}$$

The change in the distribution of the probabilities across classes is updated in the state vector at every step such that the RL agent can choose the optimum action at every step, maintaining the L_p and the number of steps/queries.

3.2. Dual-action speedup for Deep Tree Search

3.2.1 Overview and Modification to MDP

In the proposed method, the input image is divided into square patches of size $n \times n$. For a true negative case, the sensitivity of the ground truth probability (P_{GT}) to addition and removal of distortion is computed for each patch. Based on this sensitivity information, our agent takes two actions at each step: select patches to which distortions are added and selected patches to which distortions are removed. In such a case we can define the state S_t and action A_t for timestep t as:

$$S_t = S_t^+ + S_t^- \tag{4}$$

$$A_t : x \rightarrow x + \delta_t^+ - \delta_t^-, \tag{5}$$

where for timestep t, S_t^+ is the state after the add distortions perturbation δ_t^+ is performed, and S_t^- is the state after the remove distortions perturbation δ_t^- is applied.

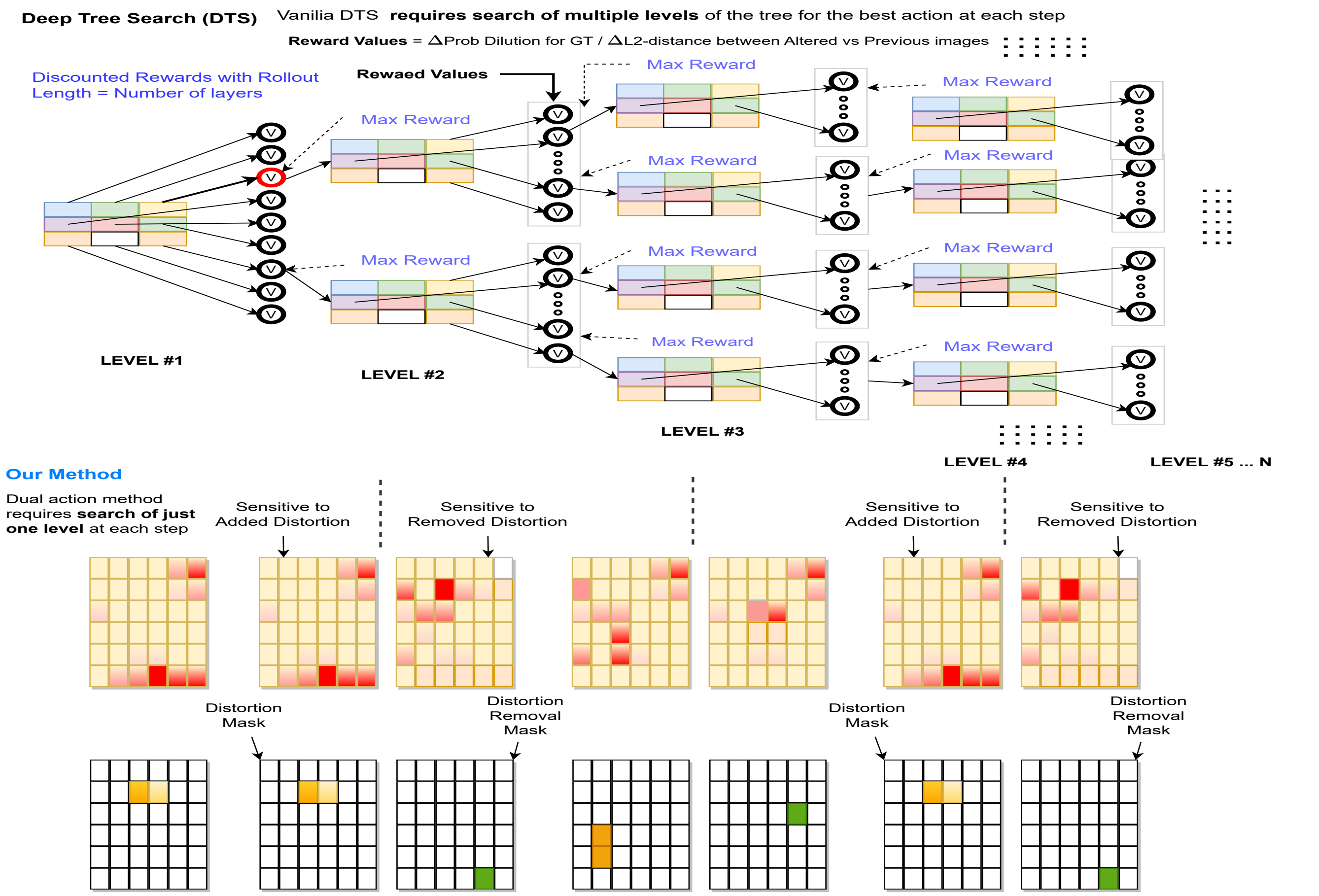


Figure 2. Dual-action architecture simplifying deep tree search

This process is iteratively performed until the model misclassifies an image or until the budget for the number of maximum allowed steps is reached. In the case of mixed filter setting, the RL agent also needs to choose the optimal type of distortion filter for each step. For introducing the distortion at different threshold levels for untargeted adversarial samples, the process continues until the threshold level of distortion is reached.

A similar technique is adopted for false positive benchmark generation with targeted adversarial samples, where the distortions are added to improve the classification probability of a non-ground truth class.

3.2.2 Intuition for dual-action

The idea of having two actions, addition, and removal, is inspired by the limitations of the RL techniques used in board games. In that setting, the most effective moves are determined through a computationally expensive process called Deep Tree Search (DTS), which looks ahead multiple layers on a longer time horizon as the game progresses. However, unlike board games, in this problem, we have the ability to undo previous moves if we realize they are suboptimal. In our framework, this is achieved by removing distortions added to patches in earlier steps and adding distortions to other patches, considering the current state of the modified image. This is similar to replaying all the moves in one step while analyzing the sensitivity of the image only at its current state, without performing a complete tree search.

By adopting this approach, we can significantly reduce the computational complexity from $O(N^d)$ to $O(N)$. Here, N represents the computation complexity of evaluating one level and corresponds to the image size, while d represents the depth of the tree search, which indicates how far ahead we look in the decision-making process.

3.2.3 Sensitivity Analysis

For the sensitivity analysis, distortion filters (masks) of size $n \times n$ are created with specific hyperparameters like distortion levels. These hyperparameters remain constant throughout the experiment. The filters are applied to square patches during training and validation to measure the change in the ground truth classification probability (P_{GT}).

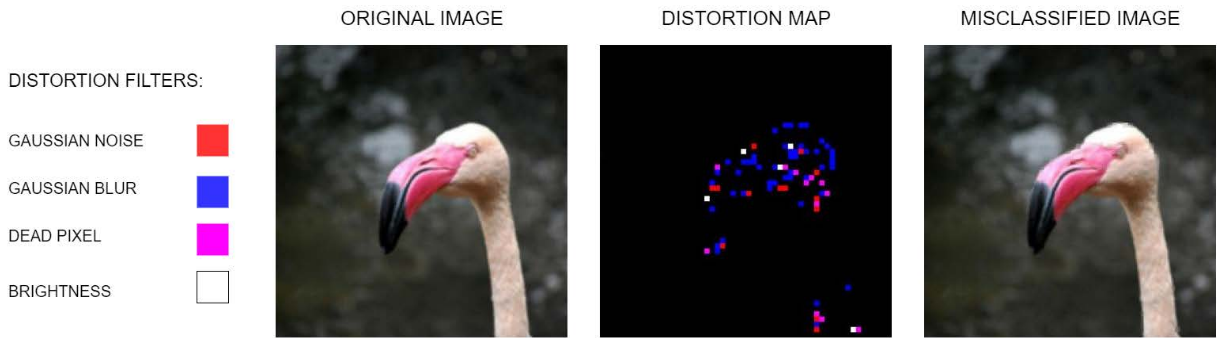


Figure 3. Mix of distortions for Adversarial Sample Generation

The hyperparameters of the distortion filters are chosen with minimal values to gradually introduce distortion and control the L_p norm effectively. The distorted samples are constrained to the range of $[0,1]^d$, where d is the dimensionality of the data. When multiple filters are available for the reinforcement learning (RL) agent to choose from, the hyperparameters are selected to have the same impact on the L_p norm after applying any filter.

3.2.4 State Vector

The state vector was designed with the output of the image sensitivity analysis ordered based on the drift in P_{GT} for patches during addition ($LIST^{+}$) and removal ($LIST^{-}$) of distortions. In addition, the classification probabilities of each class at every step ($LIST^{P}$) and the L_p norm are included in the state vector.

3.3. Flexibility to use custom distortions

Our framework offers great versatility by allowing users to apply any type of distortion of their choice. The RL algorithm within the framework learns a policy that can adapt to different filters, ensuring that adversarial samples are generated with minimal distortion, denoted as D. Additionally, the algorithm can handle a combination of filters. At each step, the agent determines which filter (e.g., Gaussian noise, Gaussian blur, brightness adjustment) to use and the number of patches to which the filter should be applied. In our experiments, we explored multiple filters and presented four naturally occurring distortion filters in this paper. Figure 1 displays adversarial examples generated using different filters, while Figure 3 showcases adversarial examples generated with a mixture of various distortion filters.

4. Metrics and Experiments

We evaluate our proposed method with two different types of distortions: Gaussian noise and Gaussian blur. Since these types of common corruptions can be subtle or destructive, we generate data with five levels of severity s and aggregate their scores. Clean error (E^{clean}) is defined as the top-1 misclassification of samples from the clean test set by evaluating the pre-existing classifier on the un-perturbed dataset. Corrupt error (E^{corrupt}) is defined as the top-1 misclassification of the samples from the corrupt dataset by evaluating the pre-existing classifier on the perturbed dataset. The performance of the classifier across the different severities levels of corruption can be represented as:

$$CE^{\text{corrupt}} = \sum_{s=1}^{5} E_s^{\text{corrupt}} \tag{6}$$

$$\text{Accuracy}^{\text{corrupt}} = 1 - CE^{\text{corrupt}} \tag{7}$$

$$CE^{\text{degradation}} = \sum_{s=1}^{5} \left(E_s^{\text{clean}} - E_s^{\text{corrupt}}\right) \tag{8}$$

Furthermore, different corruptions pose different levels of difficulty as the effect of adding Gaussian noise, Gaussian blur, and illumination do not have the same impact on the sample. Note that in our results, for better robustness, we calculate the mean across the different corruption techniques used in this work (denoted as m_{CE}). Finally, accuracy degradation is the decline in the classifier performance when evaluated on both clean and corrupted datasets.

Our benchmark is used to evaluate models from RobustBench [4], which is a reputable and continuously updated resource that both tracks and benchmarks adversarial robustness methods. The state-of-the-art models are selected by evaluating methods among thousands of papers on difficult benchmarks: L_2-constrained attacks, L_∞-constrained attacks, and corruptions on standard image classification datasets. As RobustBench has built its reputation as a core scientific resource for tracking robustness progress, we treat the best-performing methods as state-of-the-art in the literature. This is further substantiated as methods are included selectively: they cannot generally have non-zero gradients with respect to the input, have a fully deterministic forward pass, nor lack an optimization loop. It is known that the violation of these guidelines does not substantially improve robustness in general [2, 3].

4.1. Compute Details

The computation for the complete pipeline is GPU-dependent and is efficiently batched and scaled on GPUs. Caching techniques were used for pre-computed information such as the noise masks for improved efficiency. Apollo servers with 8 V100 32GB GPUs were used for training and validation, as well as the evaluation of robustness methods. We processed 16 (images per GPU) $\times$ 8 (GPUs) = 128 images in a batch for the complete pipeline.

5. Results and Discussion

5.1. CIFAR-10

To validate the effectiveness of our generated benchmark, we compare the performance of state-of-the-art robustness methods between our distorted version of CIFAR-10 [17] and CIFAR-10-C [14]. CIFAR-10-C comprises distorted versions of the CIFAR-10 test set that are applied at five different severity levels. For a fair comparison, we compute the average L_2 distance between the original test set and the CIFAR-10-C test set for each type of distortion. We then employ our framework to generate distorted versions of those data splits for the approximate average L_2 of each CIFAR-10-C severity. Due to our sample generation procedure, we do not set a target L_2 (nor do the generators of CIFAR-10-C) so we must approximate the target average L_2. In experiments, we set generation parameters empirically and keep splits that have an average L_2 of within 25%. Often, especially with Gaussian blur, our average L_2 is far lower than that of CIFAR-10-C.

We select the top-10 ranked robustness methods, which includes state-of-the-art diffusion models, on CIFAR-10-C that are reported on the RobustBench benchmark [4] for evaluation: Binary CARD(-Deck) [9], LRR CARD(-Deck) [9], AugMix-ResNeXt [15], AugMix-WRN [15], RLAT-AugMix(-JSD) [16], PRIME-ResNet18 [19], and EDM-WRN-70-16 [30]. For each severity and victim model, we generate two sets of samples with Gaussian noise and Gaussian blur distortions, respectively. We consider VGG-16, Inception-V3, and ResNet-50 as the victim models in experiments. As discussed in Section 3, our framework does not generate a sample if the victim model misclassifies it initially. Hence, we generate distorted samples on a subset of the test set. For a fair comparison, we take the same subset from both CIFAR-10 and CIFAR-10-C to compute clean and corrupted performance, respectively. This sample-wise comparison ensures that harder samples are not excluded or easier samples are not included by one split or another. This is done by storing the indices of every sample in each split, including the original split, CIFAR-10-C split, and our split to prevent samples from inflating or deflating accuracy between splits. The results of these evaluations are shown in Figure 4. For each victim model and distortion, the scores on each CIFAR-10 test set are aggregated across all five levels of severity. For the blur distortion, we cause greater or equal degradation in performance than CIFAR-10-C across all robustness methods and victim models. The except lies with EDM-WRN-70-16 on samples generated with the Inception-V3 victim model, albeit marginally. Typically, the degradation value is much higher on ours and, sometimes, over double that of CIFAR-10-C. For the noise distortion, we cause greater or equal degradation in performance than CIFAR-10-C across robustness methods and each victim model.

5.2. ImageNet

To validate the effectiveness of our generated benchmark, we also compare the performance of state-of-the-art robustness methods between our distorted version of ImageNet and ImageNet-C. Figures 5a and 5b show some examples of the images in original ImageNet, ImageNet-C, and our distorted version of ImageNet. The images shown are for severity level 5 of the Gaussian noise and blur distortions, respectively. Note that ImageNet-C comes center-cropped and thus the full images are not shown. The evaluation here is conducted in the same manner as with CIFAR-10, ensuring that noise levels are similar and that a sample-wise comparison is conducted properly. We select the top-10 ranked robustness methods, which includes state-of-the-art ViTs, on ImageNet-C that are reported on the RobustBench benchmark for evaluation: DeepAugment+AugMix [13], CondANTSpeckle-DeiT-{S,B} [28], SIN(+IN(+IN)) [11], AugMix [15], standard ResNet-50, and NoisyMix(-tuned) [10].

The results of these evaluations are shown in Figure 6. Similar to our results on CIFAR-10, our distorted version of ImageNet results in greater accuracy degradation across the robustness methods than that of ImageNet-C. Notably, the mean L_2 level of ImageNet-C (99.3) is **69.0% higher than the mean L_2 level on our distorted version of ImageNet** (58.8) for Gaussian noise for the severity level of 5. Furthermore, the mean L_2 level of ImageNet-C (79.8) is *over 3$\times$ higher than the mean L_2 level on our distorted version of ImageNet* (25.6) for Gaussian blur. In both cases, we cause *greater accuracy degradation across all robustness models*.

5.3. Results on Adversarial Retraining

Table 1 shows the retrained robustness of the target model with our framework when compared to retraining with the other competitor approaches. The table presents the degradation error percentages for image classification architectures on the CIFAR-10-C dataset, comparing state-of-the-art techniques. The degradation errors for each tech-

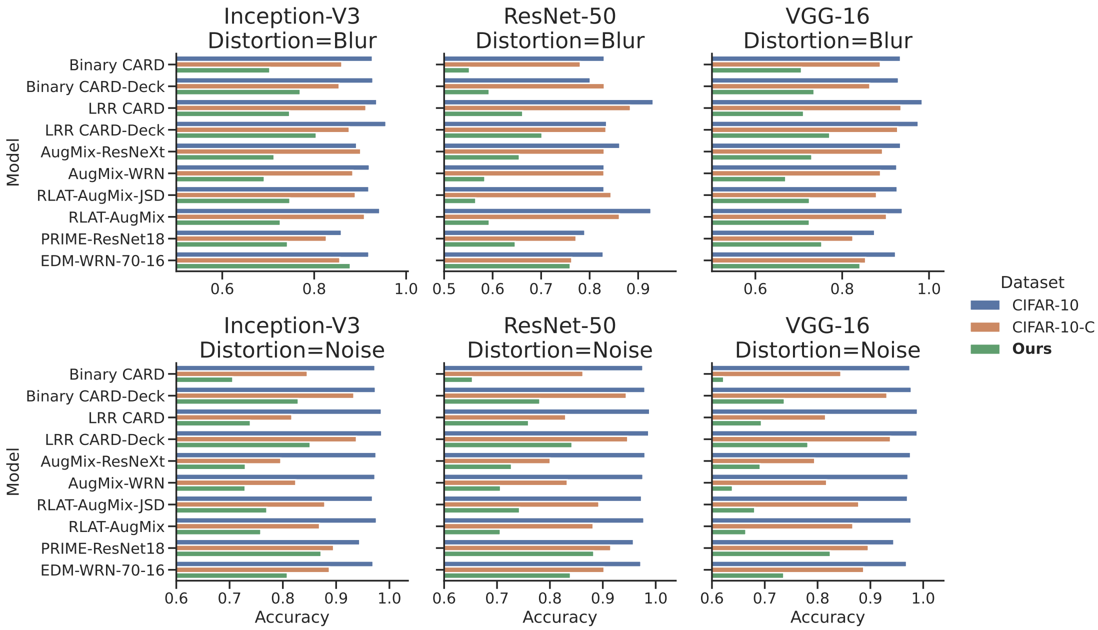
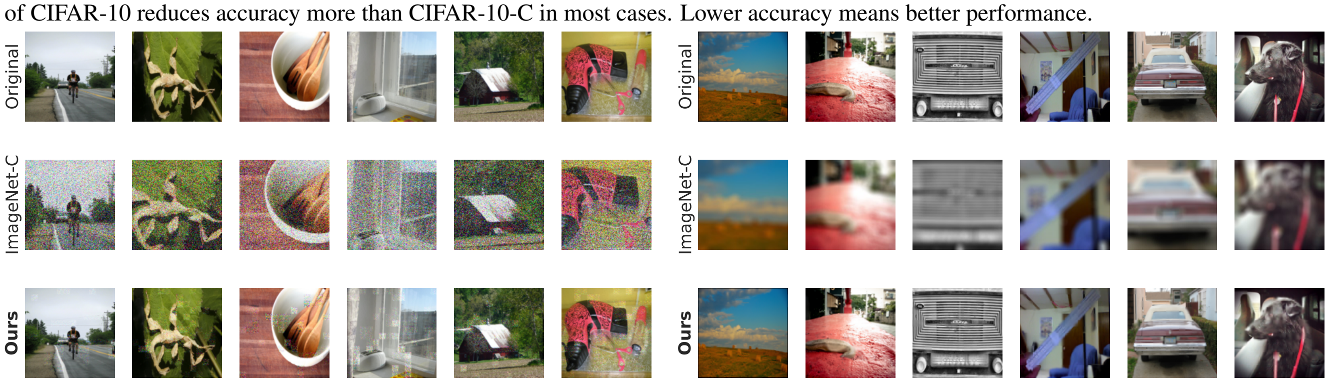


Figure 4. Evaluation of state-of-the-art robustness methods on corrupted versions of CIFAR-10: our corruptions with three victim models (ResNet-50, Inception-V3, and VGG-16) and CIFAR-10-C. Across two kinds of distortions, Gaussian noise, and blur, our corrupted version of CIFAR-10 reduces accuracy more than CIFAR-10-C in most cases. Lower accuracy means better performance.



(a) (b)

Figure 5. A subset of images from each of original ImageNet, ImageNet-C, and our distorted version of ImageNet. The images shown are for severity level 5 of the Gaussian (a) noise and (b) blur distortions. For the same severity level, images from ours retain much more clarity while being more challenging to classify.

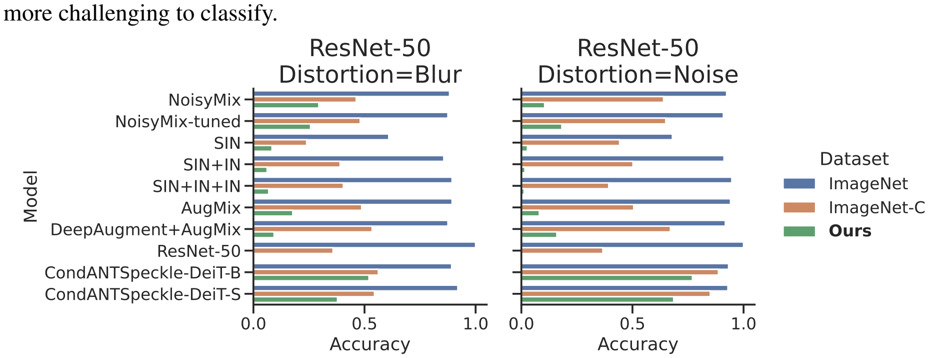


Figure 6. Evaluation of state-of-the-art robustness methods on corrupted versions of ImageNet: our corruptions and ImageNet-C. Our corrupted version of ImageNet reduces accuracy more than ImageNet-C in most cases. Lower accuracy means better performance.

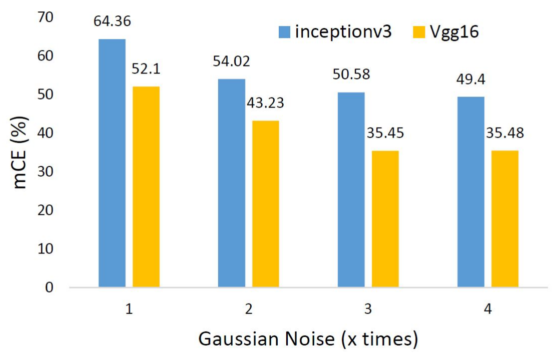


Figure 7. Evaluation of transferability of adversarial samples across other models

nique are provided for three different models: ResNet-50, DenseNet, and Inception-V3.

The results show that our framework outperforms the other techniques across all three models. For ResNet-50, we achieved a significantly lower degradation error of 6.0%, compared to Mixup (29.0%), CutMix (31.5%), and AugMix (13%). Similarly, for DenseNet and Inception-V3, our framework also demonstrates superior performance, with degradation errors of 11% and 9.5%, respectively, compared to the other techniques. These findings suggest that our framework effectively has the lowest degradation errors in image classification tasks on the CIFAR-10-C dataset, surpassing the performance of other state-of-the-art techniques like Mixup, CutMix, and AugMix.

Table 1. Degradation error % for image classification architectures on CIFAR-10-C for state-of-the-art techniques. For fairness, all of the techniques were evaluated with the same seed.

Model	Mixup	CutMix	AugMix	Ours
ResNet-50	29.0	31.5	13	**6.0**
DenseNet	24.0	33.5	15	**11**
Inception-V3	29	23	11.5	**9.5**
Mean	27.3	29.3	13.1	8.83

5.4. Transferability across different models

Table 2 represents the ability to transfer the adversarial samples across other primitive models. The adversarial samples are generated to deceive the pre-trained model shown in each row and are tested on the model shown in each column.

From the table, it can be understood that adversarial samples that were generated and evaluated on the same models have 0 accuracy. Furthermore, these adversarial samples still have a significant impact on the other primitive models showing the ability of the proposed method to generalize well. The values are averaged across both Gaussian blur and Gaussian noise types of distortions. Figure 7 illustrates the transferability of samples generated using the ResNet-50 model with the ImageNet dataset. These samples were tested on Inception-V3 and Vgg16 models under various noise levels. It can be observed that the samples generated by ResNet-50 still exhibit substantial correlation errors across different models. Also, as the noise level increases, the performance tends to decrease.

Table 2. Transferability of adversarial samples generated from CIFAR-10 across other primitive models. The values represent the classification accuracy mCE.

		ResNet-50	Inception-V3	VGG-16
Victim	**ResNet-50**	0	12.19	8.93
	Inception-V3	20.17	0	12.16
	VGG-16	16.90	16.70	0

6. Limitations

The proposed method focuses on vulnerabilities of image classifiers from distortions present at deployment by providing the customization option. Our results with the CIFAR-10-C benchmark show that our method is more effective in identifying vulnerabilities with optimal distortions that are generalizable across models. The nature of the distortion filters used by our model uncovers the broad vulnerabilities of the deployed model but does not enable unnatural artifacts.

7. Conclusion and Future Work

This paper presents a novel approach to address the challenge of evaluating and improving the robustness of neural networks against adversarial perturbations. The proposed ML-driven adversarial data generator introduces naturally occurring distortions to the original dataset, creating an adversarial subset. By formulating the problem as an MDP, the generator effectively identifies and adds distortions to the most vulnerable areas of the input. This approach demonstrates competitive performance with state-of-the-art techniques, providing a benchmark for evaluating the robustness of image classification models. Additionally, the framework allows for the inclusion of custom distortion types, adversarial thresholds, and datasets, enabling tailored evaluations and audits for specific use cases. The results highlight the importance of building robust deep-learning models and offer valuable insights for future research and development in this area. Overall, this work contributes to the advancement of reliable and resilient deep learning architectures through the generation of adversarial benchmarks and the exploration of improved adversarial training methods. In the future, we will include evaluations on additional naturally occurring perturbations.

References

[1] Sara Beery, Elijah Cole, and Arvi Gjoka. The iwildcam 2020 competition dataset. *arXiv preprint arXiv:2004.10340*, 2020. 2

[2] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019. 6

[3] Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *international conference on machine learning*, pages 1310–1320. PMLR, 2019. 6

[4] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. RobustBench: a standardized adversarial robustness benchmark. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, pages 1–17, 2021. 5, 6

[5] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pages 2206–2216. PMLR, 2020. 2

[6] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018. 2

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 2

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 2

[9] James Diffenderfer, Brian R. Bartoldson, Shreya Chaganti, Jize Zhang, and Bhavya Kailkhura. A winning hand: Compressing deep networks can improve out-of-distribution robustness. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 664–676, 2021. 2, 6

[10] N. Benjamin Erichson, Soon Hoe Lim, Francisco Utrera, Winnie Xu, Ziang Cao, and Michael W. Mahoney. Noisymix: Boosting robustness by combining data augmentations, stability training, and noise injections. *CoRR*, abs/2202.01263, 2022. 2, 6

[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 *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 2, 6

[12] Tejas Gokhale, Rushil Anirudh, Jayaraman J Thiagarajan, Bhavya Kailkhura, Chitta Baral, and Yezhou Yang. Improving diversity with adversarially learned transformations for domain generalization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 434–443, 2023. 2

[13] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 8320–8329. IEEE, 2021. 6

[14] Dan Hendrycks and Thomas G. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. 1, 2, 6

[15] 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 *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. 2, 6

[16] Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions. In James Cussens and Kun Zhang, editors, *Uncertainty in Artificial Intelligence, Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence, UAI 2022, 1-5 August 2022, Eindhoven, The Netherlands*, volume 180 of *Proceedings of Machine Learning Research*, pages 1012–1021. PMLR, 2022. 2, 6

[17] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[18] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017. 2

[19] Apostolos Modas, Rahul Rade, Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. PRIME: A few primitives can boost robustness to common corruptions. In Shai Avidan, Gabriel J. Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision - ECCV 2022 - 17th European Conference, Tel Aviv, Israel, October 23-27, 2022, Proceedings, Part XXV*, volume 13685 of *Lecture Notes in Computer Science*, pages 623–640. Springer, 2022. 2, 6

[20] Norman Mu and Justin Gilmer. Mnist-c: A robustness benchmark for computer vision. *arXiv preprint arXiv:1906.02337*, 2019. 2

[21] Raghavendran Ramakrishnan, Bhadrinath Nagabandi, Jose Eusebio, Shayok Chakraborty, Hemanth Venkateswara, and Sethuraman Panchanathan. Deep hashing network for unsupervised domain adaptation. In *Domain Adaptation in Computer Vision with Deep Learning*, pages 57–74. Springer, 2020. 2

[22] Soumyendu Sarkar, Ashwin Ramesh Babu, Vineet Gundecha, Antonio Guillen, Sajad Mousavi, Ricardo Luna, Sahand Ghorbanpour, and Avisek Naug. Rl-cam: Visual explanations for convolutional networks using reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3860–3868, 2023. 2

[23] Soumyendu Sarkar, Ashwin Ramesh Babu, Vineet Gundecha, Antonio Guillen, Sajad Mousavi, Ricardo Luna, Sahand Ghorbanpour, and Avisek Naug. Robustness with query-efficient adversarial attack using reinforcement learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2329–2336, 2023. 2

[24] Soumyendu Sarkar, Ashwin Ramesh Babu, Sajad Mousavi, Sahand Ghorbanpour, Vineet Gundecha, Ricardo Luna Gutierrez, Antonio Guillen, and Avisek Naug. Reinforcement learning based black-box adversarial attack for robustness improvement. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pages 1–8. IEEE, 2023. 2

[25] Soumyendu Sarkar, Ashwin Ramesh Babu, Sajad Mousavi, Vineet Gundecha, Sahand Ghorbanpour, Alexander Shmakov, Ricardo Luna Gutierrez, Antonio Guillen, and Avisek Naug. Robustness with black-box adversarial attack using reinforcement learning. In *AAAI 2023: Proceedings of the Workshop on Artificial Intelligence Safety 2023 (SafeAI 2023)*, volume 3381. https://ceur-ws.org/Vol-3381/8.pdf, 2023. 2

[26] Soumyendu Sarkar, Sajad Mousavi, Ashwin Ramesh Babu, Vineet Gundecha, Sahand Ghorbanpour, and Alexander K Shmakov. Measuring robustness with black-box adversarial attack using reinforcement learning. In *NeurIPS ML Safety Workshop*, 2022. 2

[27] Jiri Sedlar, Karla Stepanova, Radoslav Skoviera, Jan K Behrens, Matus Tuna, Gabriela Sejnova, Josef Sivic, and Robert Babuska. Imitrob: Imitation learning dataset for training and evaluating 6d object pose estimators. *IEEE Robotics and Automation Letters*, 8(5):2788–2795, 2023. 2

[28] Rui Tian, Zuxuan Wu, Qi Dai, Han Hu, and Yugang Jiang. Deeper insights into vits robustness towards common corruptions. *arXiv preprint arXiv:2204.12143*, 2022. 2, 6

[29] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR, 2019. 2

[30] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. *CoRR*, abs/2302.04638, 2023. 2, 6

[31] Zhenlin Xu, Deyi Liu, Junlin Yang, Colin Raffel, and Marc Niethammer. Robust and generalizable visual representation learning via random convolutions. *arXiv preprint arXiv:2007.13003*, 2020. 2

[32] 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/CVF international conference on computer vision*, pages 6023–6032, 2019. 2

[33] Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. 2