Don't Discriminate Among Corruptions: Generative Robustness Network for Unified Resiliency

Anonymous authors
Paper under double-blind review

Abstract

The vulnerability of deep recognition algorithms against image degradations widens the gap between their performance as compared to robust human perception. Such degradations can be intentionally crafted or commonly occur. While both types of corruption have a similar adverse impact, they are handled independently in the existing literature. We assert they should not be dealt with independently if we aim to build a universally secure and robust system. In this research, we address both types of image degradations (referred to as common corruptions and intentional adversarial perturbations) using a single unified framework: *generative robustness network*. Using the proposed framework, we present: (i) the detection of degraded samples, (ii) mitigating the impact of these degradations, and (iii) establishing a possible connection between different types of degradations. This research presents a *'universal'* framework using a generative robustness network to safeguard deep learning algorithms against both common and deliberate corruptions. We assert that not only the degradations but also their severity is critical since they can widen *intra* and *inter* class variations. Extensive experiments are performed using multiple datasets and degradations in both seen and unseen settings to demonstrate the efficacy of the proposed framework.

1 Introduction

It is well documented that machine learning algorithms are not robust to common corruptions and adversarial perturbations unlike human perception Azulay & Weiss (2018); Recht et al. (2018); Yin et al. (2019). Common image corruptions can occur naturally due to several environmental factors such as imaging sensor, viewpoint, and illumination; whereas, adversarial perturbations are intentional modifications of the images designed to fool classification systems. Common corruptions are generally perceptible while adversarial perturbations are designed to be quasi-imperceptible. While the perceptibility of common corruptions is an issue (from an attack effectiveness standpoint), they do not require any knowledge of the target classifier to be effective which is an advantage. In contrast, the majority of the adversarial perturbations require access to the target machine learning classifiers, whether it be their exact architecture/weights or score probabilities. Through proper adjustment of the strength/severity of the common corruptions, they can also be quasi-imperceptible and still fool the classifiers.

Even though these degradations originate from two completely different mechanisms, distinguishing the added noise patterns might be near-impossible through observation. Fig. 1 presents a case for not discriminating between natural corruptions and adversarial perturbations while designing a defense mechanism. The noise patterns shown in the image are extracted from five images in the CIFAR-10 dataset Krizhevsky et al. (2009) and correspond to two different types of common corruptions and two different adversarial perturbation generation algorithms. It is difficult to distinguish which pattern belongs to which category by merely looking at the noise patterns. Referring to Fig. 1, the patterns shown in A and D correspond to natural corruptions: Gaussian noise and Uniform noise, respectively. Whereas, the noise patterns B and C are deliberate adversarial perturbations: iterative Fast Gradient Sign Method (IFGSM) Kurakin et al. (2018) and Projected Gradient Descent (PGD) Madry et al. (2017). Fig. 2 shows the images that are modified using the corruptions depicted in Fig. 1. While the images shown in B contain visually perceptible

Figure 1: Four different image degradation patterns are obtained using the images of the CIFAR-10 dataset. The degradations are broadly categorized as common corruptions and adversarial perturbations. Can you identify which degradation patterns belong to which category? Each noise pattern is amplified by a factor of 10 for better visibility. Hint: Each row is one degradation type.

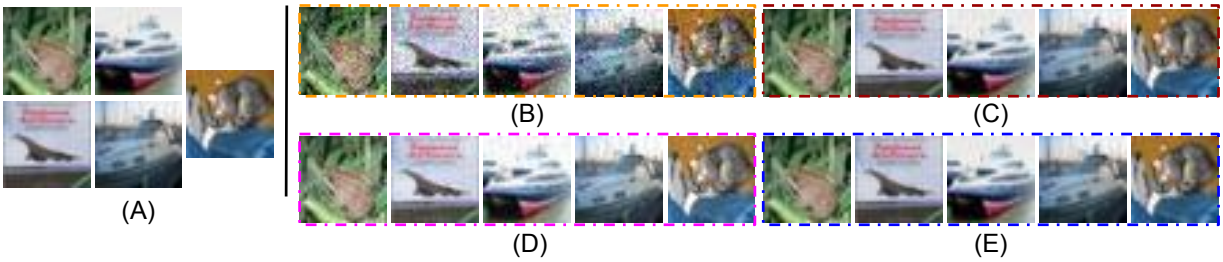

Figure 2: The noisy images generated using the common corruptions and adversarial perturbations. A represents the five different clean images. B, C, D, and E are their degraded versions.

modifications, human observers can still classify them correctly with relative ease and confidence. Another common corrupted noisy image shown in D contains quasi-imperceptible modifications similar to the adversarial perturbations shown in C and E. The examples further motivate how not discriminating between these degradations can help build a unified robust machine-learning framework.

In the literature, several defenses against adversarial perturbations are proposed which vary from adversarial training to the development of perturbation detectors Abusnaina et al. (2021); Andriushchenko & Flammarion (2020); Deng et al. (2021); Serban et al. (2020); Shafahi et al. (2020). However, the literature of defenses towards common corruptions is shallow. Popular defenses against common corruptions are based on large-scale data augmentation or utilizing a heavy network backbone Geirhos et al. (2018); Hendrycks et al. (2019); Mahajan et al. (2018); Rusak et al. (2020); Xie et al. (2020). Existing adversarial training and data augmentation methods are computationally expensive and are not found to be generalizable against unseen corruptions or perturbations with acceptable efficacy Calian et al. (2021); Mintun et al. (2021). Recently, a few works have started framing defenses from the perspective of denoising Xie et al. (2019); Salman et al. (2020). The significant drawback of existing defenses is that these degradations are dealt with independently and where common corruptions have received due attention too Hendrycks et al. (2019); Rusak et al. (2020); Serban et al. (2020); Zhang & Li (2019). This research aims to bridge that gap to build a trustworthy machine-learning framework that is robust against both types of corruption. To summarize, the contributions of this research are as follows: ❶ a unified corruption detection and mitigation framework termed as generative robustness network is proposed which can handle both common corruptions and adversarial perturbations effectively, ❷ a novel image noise remover architecture has been proposed to mitigate the impact of degradations and preserve CNN performance, and ❸ connections between natural corruptions and adversarial perturbations are identified from both detection and mitigation perspectives. These can bolster future research for resilience towards seen and unseen perturbations of both types.

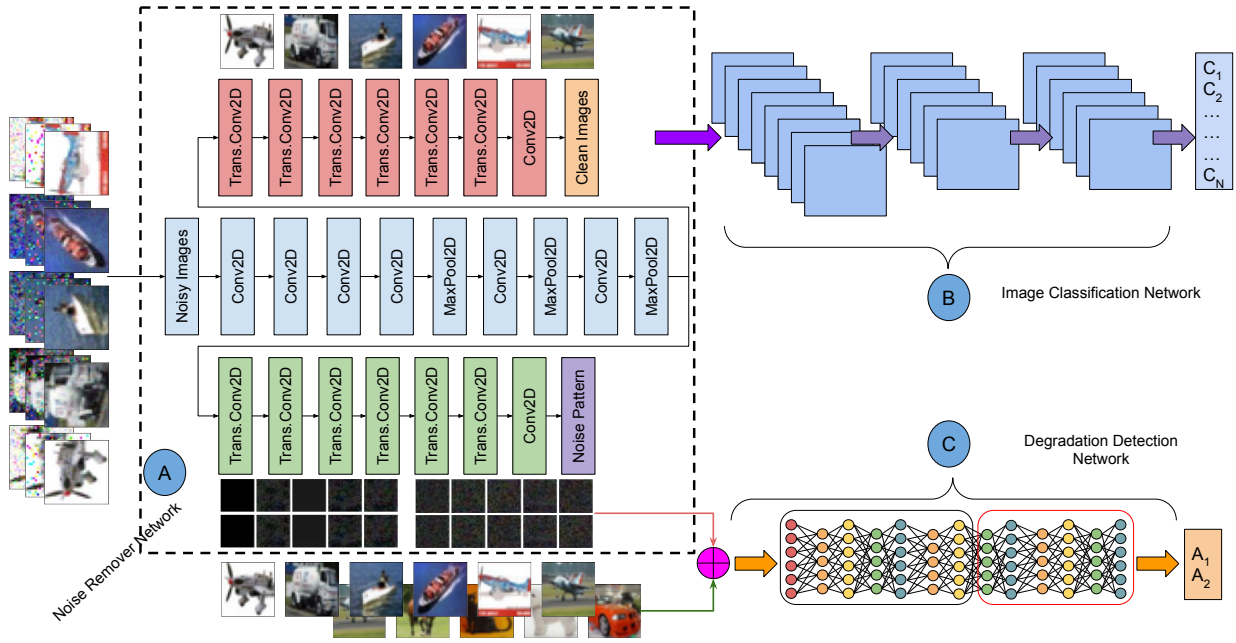

Figure 3: The proposed framework contains three high-level blocks: (A) aims to remove the noise from the images by breaking the images into two parts: the clean image and noise patterns, and (B) image classification network which utilizes the clean images generated from the noise remover block for enhanced image classification, and (C) degradation detection network aiming to classify the images into binary classes (A_1: clean or A_2: corrupted). $C_1, C_2, \cdots, C_N$ represent the N classes of the problem. $\bigoplus$ represents data augmentation along the channel dimension where the clean image and noise pattern are concatenated.

2 Proposed Unified Security Framework for Degradation Detection and Mitigation

An overview of the proposed framework is presented in Fig. 3. To accomplish our objective of handling both the detection and mitigation of corruption, we use a multi-output encoder network combined with two decoder networks Mao et al. (2016). One of the decoder networks tries to obtain a clean image without any noise or corruption and the other tries to segregate the noise pattern itself. Then, the decoded clean image is sent to the classifier network to obtain the class label for the image whereas the noise pattern obtained by the other decoder network acts as auxiliary information for degradation detection. These noise-added images can be used in conjunction with their clean versions to train a detection network that forms the final component of the framework and can distinguish between clean and corrupted images. This enables a modular structure so that the framework can accommodate different choices of independent networks that can fulfill the defined roles while being well-suited to the problem at hand. For instance, applications dealing with complex images may use a deeper classification network whereas others may be able to save computational resources by substituting a lightweight classifier with fewer parameters instead Baldock et al. (2021); Jiang et al. (2021).

2.1 Training

The entire training framework of the proposed trustworthy machine learning system can be broken down into three pieces: (i) in the first piece a noise remover network is trained to utilize the degraded images, (ii) an image classification network is trained on the number of given classes in the dataset, and (iii) an image degradation network is trained to segregate the corrupted data from the clean data points. While each component can be trained independently, in the inference time they are somewhat interconnected.

To train the noise remover network a deep encoder-decoder is utilized which takes the noisy data as input. To achieve the noisy images, we leverage a set of clean images, say $F(x, y)$, and obtain a corrupted version of each image in the set as follows:

$$g(x, y) = f(x, y) + \eta(x, y) \tag{1}$$

Here, $\eta(x, y)$ denotes the noise pattern such as Gaussian or Speckle added to the clean image $f(x, y) \in F(x, y)$ and corrupted image $g(x, y)$ is obtained. Then, the autoencoder network (A in Fig. 3) is trained using the set of corrupted images $g(x, y)$ as the input. For each input, the objective of the proposed autoencoder network is to segregate the image into a clean image and noise pattern, such that the loss of the network is defined as:

$$\mu(f'(x, y), f(x, y)) + \mu(\eta'(x, y), \eta(x, y)) \tag{2}$$

Here, μ denotes the mean square error function, and $f'(x, y)$ and $\eta'(x, y)$ denote the output of the two decoder networks, i.e., the reconstructed clean image and the reconstructed noise pattern, respectively. The advantage of segregating the clean images and noise patterns separately comes from the belief that each noise generation method impacts different regions of an image due to the varying strength of the noise vector. Therefore, a simple mapping of noisy images to their clean counterpart cannot properly handle that variation in noise type.

Next, we train the degradation detection network (C in Fig. 3) using $f(x, y)$ and $g(x, y)$ for binary classification. However, in place of providing images of these two classes, the images are passed through the trained noise remover network to obtain the possible noise pattern added to the images. In the case of images, we assert that no blank pattern will be outputted by the network. Whereas, the noisy images will output a significant noise vector. Let $\eta'_f(x, y)$ and $\eta'_g(x, y)$ are the obtained noise vectors from the clean and noisy images, respectively. We concatenated them along the channel dimension into the respective images. If the dimensions of the original image are denoted by $M * N * c$ where M is the width, N is the height, and c is the number of channels, then the input to the detection network has the dimension $M * N * 2c$. In this manner, the detection network can learn to utilize the mapping between the noise pattern of a corrupted image and the absence of noise pattern obtained for a clean image to classify the input as clean (A_1) or corrupted (A_2). The objective function of the decoder network is to maximize the binary class accuracy.

The final component in the proposed framework is the classifier network (B in Fig. 3) that can be trained separately and independently as it can presume that clean input will be provided to it ($f(x, y)$). It is only concerned with maximizing the class-wise accuracy or other metrics that would typically be used for the problem domain in the absence of corruption. This differentiates the proposed framework from many other adversarial training-based defenses where the classifier network itself has to be trained on both clean and corrupt samples and as a consequence, the defense does not perform well on unseen corruption types and is computationally expensive. Geirhos et al. (2018); Kim et al. (2020); Korkmaz (2021); Mintun et al. (2021); Zhang et al. (2019).

2.2 Inference

During inference, an image $h(x, y)$ is provided to the noise remover network first which generates a reconstructed clean image $h'(x, y)$ and a reconstructed noise pattern $\eta'_h(x, y)$. The clean image $h'(x, y)$ is used for classification by the classifier network to obtain the class label prediction $(C_1, C_2, ..., C_n)$. The degradation detection network is provided the concatenated image $h'(x, y) \bigoplus \eta'_h(x, y)$ as input that can then be classified as clean (A_1) or corrupted (A_2). The proposed framework is referred to as a *'generative robustness network'* (GRN) where generative refers to yielding clean and noise patterns and robustness corresponds to enhanced classification accuracy and degradation detection.

2.3 Implementation Details

The noise remover encoder-decoder contains six conv layers containing 64, 128, and 256 filters in a pair of layers. ReLU activation is applied at each layer except the last layers utilizing sigmoid activation. The classifier architecture can be any such as on the CIFAR datasets we have used ResNet He et al. (2016) and Wide-ResNet Zagoruyko & Komodakis (2016). For image degradation detection, we have used the Wide-ResNet10-2 architecture. Each component of the framework is trained end-to-end without any pre-training.

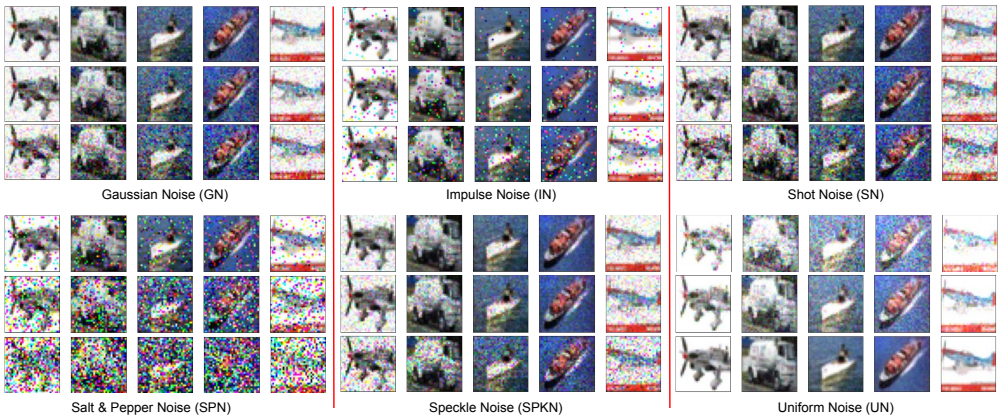

Figure 4: Images imbued with common corruptions with different severity levels mentioned in Table 1 which also changes their perceptibility from perceptible to quasi-imperceptible.

Table 1: Parameters used for the different severity levels of each common corruption.

Severity	Gaussian Noise (GN)	Uniform Noise (UN)	Salt & Pepper Noise (SPN)	Shot Noise (SN)	Impulse Noise (IN)	Speckle Noise (SPKN)
1	0.08	0.5	0.1	60	0.03	0.15
2	0.12	0.3	0.3	25	0.06	0.2
3	0.18	0.1	0.5	12	0.09	0.35

Computational Cost: The proposed architecture aiming to provide detection and mitigation security against several image degradations is a computationally efficient paradigm as well. The proposed noise remover and detection network took a total of 25 minutes on a single GPU NVIDIA GeForce RTX 2080 machine. It makes the proposed algorithm ready for real-world deployment.

3 Experimental Results and Analysis

In this research, we have conducted an extensive experimental analysis using multiple popular object recognition datasets, namely CIFAR-10 Krizhevsky et al. (2009), CIFAR-100 Krizhevsky et al. (2009), Tiny ImageNet Le & Yang (2015), and a subset of ImageNet Deng et al. (2009). The CIFAR datasets contain pre-defined train and test sets of $50,000$ and $10,000$ images, respectively. We have used six common corruptions and generated the noisy images on these datasets with varying levels of severity of the corruptions. The variations in corruption along with their severity help in understanding the impact of the proposed research and establishing its strength in building the universal security framework. The scale parameters of each corruption: Gaussian noise (GN), Uniform noise (UN), Salt and pepper noise (SPN), Shot noise (SN), Impulse noise (IN), and Speckle noise (SPKN) are detailed in Table 1. The corrupted images generated using varying severity levels are also presented in Fig. 4.

3.1 Detection

In the first study, we performed experiments for the detection of common corruption-induced images on the CIFAR datasets. Corrupted images are generated for both the training and testing sets; the training set corrupted and clean images are used for learning the noise remover and detector networks whereas the testing set is used for evaluation. The results cover both seen and unseen scenarios including seen severity levels and unseen severity levels even across the corruption types. Extensive results are important to estimate unconstrained real-world performance where the defended network might not have been trained on the same type of corruption encountered at the time of testing or its severity level might be different as compared to the severity level used for training.

Table 2: Common corruption detection performance on the CIFAR-10 dataset. The proposed detection network shows high robustness in handling multiple severities effectively. The lowest severity corruption trained detector generalizes well against seen and unseen corruption. The detailed interpretative caption is given in the text as well.

Train ↓	GN			UN			SPN			SN			IN			SPKN		
Test →	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
GN-1	99.97	99.97	99.97	98.47	99.12	92.9	99.95	86.24	61.85	99.87	99.88	99.83	99.69	99.95	99.96	99.1	99.41	99.34
GN-2	99.98	99.98	99.98	99.79	99.59	69.00	99.98	99.98	99.98	99.92	99.97	99.98	99.98	99.98	99.98	99.4	99.8	99.95
GN-3	87.34	99.97	100	96.93	87.48	50.00	100	100	100	91.95	99.79	99.94	86.46	100	100	66.36	90.39	99.53
UN-1	96.86	99.83	99.99	99.99	99.92	50.33	99.99	99.99	99.99	92.98	98.47	99.52	93.77	99.85	99.99	82.25	90.7	97.5
UN-2	99.98	99.98	99.98	99.98	99.98	89.09	99.98	99.98	99.98	99.94	99.96	99.97	99.97	99.98	99.98	99.59	99.88	99.94
UN-3	99.89	99.89	99.89	98.98	99.63	99.88	99.88	99.89	99.89	99.66	99.69	99.65	61.29	99.02	99.89	98.39	98.51	98.75
SPN-1	96.33	99.99	100	98.35	94.28	50.0	100	100	100	95.49	99.84	99.95	98.75	100	100	77.9	94.21	99.73
SPN-2	52.28	93.09	99.77	88.18	54.57	50.0	99.99	100	100	58.21	95.53	99.61	58.29	99.44	99.98	50.87	69.92	97.14
SPN-3	50.00	52.60	89.56	63.53	50.01	50.0	97.91	100	100	50.08	61.33	91.25	50.01	56.53	93.64	50	50.91	80.88
SN-1	99.99	99.99	99.99	98.86	98.55	63.67	99.99	99.99	99.99	99.99	99.99	99.99	99.05	99.99	99.99	99.99	99.99	99.99
SN-2	99.6	99.99	99.99	97.17	95.83	50.39	99.99	99.99	99.99	99.98	99.99	99.99	98.52	99.99	99.99	99.91	99.98	99.99
SN-3	95.64	99.95	100	98.11	91.35	50.03	100	100	100	99.29	99.99	100	96.81	99.99	100	84.01	99.53	100
IN-1	99.99	99.99	99.99	99.96	99.79	84.54	99.99	99.99	99.99	99.97	99.99	99.99	99.99	99.99	99.99	99.67	99.9	99.98
IN-2	100	100	100	99.91	99.63	54.91	100	100	100	99.88	99.98	99.99	100	100	100	98.54	99.7	99.95
IN-3	98.81	99.99	100	98.01	95.57	50.02	100	100	100	98.84	99.9	99.97	99.73	100	100	88.64	97.24	99.8
SPKN-1	99.96	99.96	99.96	99.04	99.04	88.93	99.96	99.96	99.96	99.96	99.96	99.96	99.96	99.96	99.96	99.96	99.96	99.96
SPKN-2	99.97	99.99	99.99	98.22	97.81	58.6	99.99	99.99	99.99	99.99	99.99	99.99	99.61	99.99	99.99	99.99	99.99	99.99
SPKN-3	98.47	99.95	99.99	97.95	94.49	50.43	99.99	99.99	99.99	99.8	99.99	99.99	99.66	99.99	99.99	99.78	99.99	99.99

3.1.1 Seen Distribution Detection

The results of common corruption detection on the CIFAR-10 and CIFAR-100 datasets are reported in Table 2 and Table 3, respectively. *Each row contains results when the framework is trained on the particular noise with a specific severity level as indicated and then tested on images with noise about all the noise and severity level combinations across each column.* The lowest performance when trained with severity level 1 noise is highlighted in color for a quick understanding of the strength of each corruption in detection.

The analysis here is divided into corruption type and correction severity. Uniform noise with severity three is proven effective in handling each type of corruption (both seen and unseen) and each severity level of the corruption. When trained using corrupted images generated using UN-3, the detection network yields at least 98.39% detection accuracy except for 61.29% on the testing images with a severity one of the impulse noise (IN-3). It might be due to the low level of severity or imperceptibility of corruption. The phenomenon is observed in each corruption where training with the corruption of low-level severity generalizes well towards the same corruption, unseen corruption, same level of severity, and unseen severity level. Interestingly, the uniform noise with severity three is found complex to be detected by the majority of the noise corruptions. For instance, SPN gives random accuracy on this low level of uniform noise, an approximately random chance accuracy is observed when the detection algorithm is trained on each corruption of shot noise (SN). Moreover, the uniform noise itself when trained with high severity (SV1) is found ineffective in handling the low-level uniform noise. However, a few noises such as GN, IN, and SPKN with the lowest level of severity used in this research showcase their effectiveness in identifying the uniform noise images of low severity. The SPKN with a severity one even surpasses the lowest detection accuracy of UN-3 and yields at least 88.93% detection accuracy.

Similar behavior can be observed on the CIFAR-100 dataset where the SPKN-1 shows the highest level of performance by achieving the highest detection accuracy across the range of corruptions. Although the problem is still a two-class classification; however, the corruption detection performance suffers significantly when moved from CIFAR-10 to the CIFAR-100 dataset. For example, on the CIFAR-10 dataset, the lowest performance achieved by SPKN-1 is 88.93% as compared to the lowest performance (67.93%) on the CIFAR-100 dataset. In contrast, the Gaussian noise with severity one which shows the lower generalizability against the SPN with severity 3 (61.85%) on the CIFAR-10 dataset does not show any vulnerability in handling the SPN noise on the CIFAR-100 dataset. Overall, Gaussian noise and SPKN noise show a higher level of efficiency in the detection of common corruption which is extensively tested in several generalized settings.

Table 3: Common corruption detection performance on the CIFAR-100 dataset. Common corruption detection performance on the CIFAR-10 dataset. The proposed detection network shows high robustness in handling multiple severities effectively. The lowest severity corruption trained detector generalizes well against seen and unseen corruption.

Train ↓	GN			UN			SPN			SN			IN			SPKN		
Test →	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
GN-1	99.98	99.98	99.98	98.28	97.92	66.43	99.98	99.98	99.98	99.49	99.79	99.88	99.87	99.98	99.98	95.9	98.09	99.37
GN-2	99.94	99.99	99.99	97.18	96.01	53.28	99.99	99.99	99.99	99.63	99.89	99.95	99.51	99.99	99.99	97.56	99.14	99.73
GN-3	98.69	99.98	99.98	99.24	97.68	50.1	99.98	99.98	99.98	95.77	99.33	99.75	98.25	99.98	99.98	85.47	93.82	98.86
UN-1	89.45	97.32	99.38	99.97	99.56	50.76	99.11	97.76	99.08	87.34	95.25	97.54	89.89	97.82	98.92	78.39	87.33	95.23
UN-2	99.97	99.99	99.98	99.99	99.99	76.67	99.4	98.14	99.8	98.91	99.37	99.33	84.67	98.15	99.23	94.47	96.45	97.51
UN-3	99.83	99.8	98.18	96.48	99.47	99.81	74.25	69.64	52.24	99.55	99.12	93.59	55.25	69.08	73.79	98.01	97.29	87.45
SPN-1	81.0	97.3	99.99	91.96	78.11	50.01	100	99.98	100	82.58	98.49	99.62	86.58	99.98	100	66.9	86.13	98.05
SPN-2	70.4	97.07	99.93	95.81	80.92	50.01	99.98	99.8	99.98	78.26	97.32	99.66	86.44	99.85	99.98	67.19	84.82	98.23
SPN-3	50.02	55.31	90.29	68.26	50.01	50	98.07	62.24	100	50.04	66.05	91.68	50.03	62.05	95.35	50.01	50.82	82.50
SN-1	99.98	99.98	99.98	95.67	96.63	75.84	99.91	96.79	99.98	99.98	99.98	99.98	64.7	96.72	99.69	99.97	99.98	99.84
SN-2	99.95	99.95	99.95	99.33	98.59	77.6	99.95	99.95	99.95	99.95	99.95	99.95	99.94	99.95	99.95	99.94	99.95	99.95
SN-3	99.51	99.98	99.98	98	95.53	51.11	99.98	99.98	99.98	99.98	99.98	99.98	99.92	99.98	99.98	99.35	99.95	99.98
IN-1	99.96	99.99	99.99	99.51	98.26	51.79	99.99	99.99	99.99	99.3	99.92	99.98	99.99	99.99	99.99	96.1	98.54	99.80
IN-2	99.54	99.98	99.98	99.12	96.85	50.06	99.98	99.98	99.98	97.9	99.73	99.96	99.98	99.98	99.98	90.68	97.03	99.60
IN-3	75.58	98.62	100	94.06	82.29	50.01	100	100	100	77.58	96.93	99.33	86.86	100	100	65.14	79.45	99.96
SPKN-1	99.93	99.97	99.97	93.11	94.51	68.8	99.85	96.47	99.97	99.97	99.97	99.97	67.93	96.49	99.7	99.97	99.97	99.97
SPKN-2	99.9	99.9	99.9	99.55	99.26	93.17	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.9	99.90
SPKN-3	97.67	99.98	99.98	96.52	94.02	50.56	99.98	99.98	99.98	99.98	99.98	99.98	99.89	99.98	99.98	99.92	99.98	99.98

Table 4: Common corruption detection in the out-of-distribution (unseen/cross dataset) scenario. Similar to the seen dataset protocols, the proposed detection network can handle out-of-distribution images whether coming from the same or different corruptions with high efficacy.

Train ↓	GN	UN	SPN	SN	IN	SN	GN	UN	SPN	SN	IN	SN
Test →	Train CIFAR-10 → Test CIFAR-100						Train CIFAR-100 → Test CIFAR-10					
GN	99.9	90.89	99.84	99.66	99.23	98.4	99.99	67.02	99.99	99.73	99.95	96.78
UN	99.47	99.46	99.47	99.13	61.5	97.62	99.9	99.89	74.69	99.73	55.29	98.57
SPN	93.59	50.01	99.98	93.41	98.55	76.34	83.81	50.0	100	86.62	87.34	68.67
SN	99.91	64.2	99.91	99.91	98.82	99.91	100	77.85	99.92	100	65.1	99.99
IN	99.87	82.62	99.87	99.79	99.87	99.01	99.99	52.28	100	99.74	100	97.61
SPKN	99.73	87.11	99.73	99.73	99.73	99.73	99.95	68.8	99.89	99.99	68.42	99.99

The above results reveal that there might be some connection that can be utilized to detect common corruptions in the out-of-distribution images. Out-of-distribution is an interesting scenario Mandal et al. (2019); Ren et al. (2019); Vyas et al. (2018) where the images of one dataset are used for training and images of other datasets are used for evaluation. However, such a study is not explored for the detection of corruption examples in unseen or out-of-distribution samples. In the case of the proposed research, the importance of such a study can be thought from the following points: the datasets used for corruption belong to object recognition and are of the same resolution (32×32), and in the case of zero-shot setting, images of unseen object class might come which might be corrupted or not. However, the decision of whether they belong to any corrupted category is critical for the right decision.

3.1.2 Out-of-Distribution Detection

Towards out-of-distribution corruption examples detection, two-way experiments are performed: (i) training on the CIFAR-10 dataset using each corruption individually and tested on the images of the CIFAR-100 dataset corrupted using seen or unseen corruption of lowest severity used and (ii) the reverse case. When the corruption detection algorithm is trained on the CIFAR-10 dataset and tested on the CIFAR-100 dataset, except in a few cases, the detection algorithms show significantly higher performance (Table 4). The few cases where the performance is lacking are when the detector is trained on the UN corruption and tested on the IN corruption. In another scenario, training on SN and testing on unseen UN attacks shows a low detection performance (64.2%). We want to highlight here that in these cases not only the datasets are

Table 5: Adversarial example detection on the CIFAR-10 and CIFAR-100 datasets when the detectors are trained on each common corruption of varying severity. Colored boxes highlight the best performing common corruptions in identifying adversarial examples.

Train →	GN			UN			SPN			SN			IN			SPKN		
Test ↓	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Evaluation on the CIFAR-10 dataset																		
IFGSM	50.06	50.01	50.00	50.01	50.07	82.86	50.00	50.00	50.00	50.05	50.01	50.00	50.05	50.01	50.0	50.81	50.05	50.00
PGD	50.43	50.02	49.97	49.97	50.24	99.46	49.98	49.97	50.00	50.42	49.94	49.97	50.08	49.97	49.96	58.24	50.41	49.96
Evaluation on the CIFAR-100 dataset																		
IFGSM	50.02	50.00	50.00	50.08	50.04	87.23	50.00	50.0	50.0	50.16	50.1	50.02	50.0-	50.01	50.01	50.12	50.91	50.02
PGD	50.15	50.00	50.01	50.12	50.12	99.49	50.01	50.00	50.00	52.6	51.95	50.14	50.02	50.01	50.01	51.53	61.04	50.02

Table 6: Common corruption detection on the CIFAR-10 and CIFAR-100 datasets where the detectors are trained on adversarial perturbations. In contrast to adversarial examples detection trained on common corruption, adversarial detector shows significant success in identifying common corruption. Colored boxes highlight the strongest common corruption (yielding the lowest accuracy) to be defended by the detectors.

Test →	GN			UN			SPN			SN			IN			SPKN		
Train ↓	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Evaluation on the CIFAR-10 dataset																		
FGSM40	97.65	97.65	97.65	96.86	97.11	97.43	97.65	97.65	97.65	97.65	97.3	97.62	87.02	97.5	97.65	97.5	97.47	97.5
PGD40	99.87	99.73	99.42	95.06	97.96	99.26	92.01	97.41	97.83	99.52	98.76	97.69	54.45	79.04	90.31	97.02	95.66	90.71
Evaluation on the CIFAR-100 dataset																		
IFGSM40	99.61	99.6	99.61	94.91	95.52	97.25	99.55	95.96	99.61	99.45	99.45	99.39	60.33	95.99	99.44	98.54	98.58	98.7
PGD20	99.6	99.6	99.6	93.27	94.88	96.56	98.37	86.13	99.6	99.42	99.3	99.19	53.55	86.3	97.2	98.26	98.11	97.42

different but the corruption type is also different and it reveals the high inter-class variability among the corruptions. Gaussian noise (GN) shows the highest level of generalization both towards out-of-distribution data but also towards unseen corruption and yields at least 90.89% accuracy on the CIFAR-100 images. Speckle noise (SPKN) was found second-best in identifying the corrupted images in such wide variations in testing protocol. The effectiveness of GN corruption is also seen under the reverse case, i.e., where the training has been done on CIFAR-100 and testing is performed on the CIFAR-10 dataset.

3.1.3 Open-set Adversarial Perturbation Detection

Adversarial perturbations have tremendous success in fooling 'any' deep neural network classifier by being quasi-imperceptible Agarwal et al. (2020); Hendrycks et al. (2021); Pedraza et al. (2021). Several research works have been proposed to tackle the detection of adversarial examples in generalized settings Abusnaina et al. (2021); Agarwal et al. (2021a;b); Drenkow et al. (2022); Goswami et al. (2019); however, they are not explored in the context of their connection with common corruptions. Both kinds of degradation have proven effective in fooling CNNs with high confidence. Therefore, to build a universally secure system we tackle both degradations simultaneously. *The first question we ask is whether a detector trained on common corruptions can be effective in detecting adversarial perturbations.* If yes, then is each corruption equally effective for this purpose or does any specific corruption perform better in identifying adversarial examples? We have generated two highly effective and complex adversarial perturbations namely iterative FGSM (IFGSM) Kurakin et al. (2018) and PGD Madry et al. (2017). Interestingly, almost all common corruptions show ineffectiveness in detecting adversarial examples for both datasets. However, surprisingly, uniform noise (UN) with the lowest severity shows a very strong correlation between the image degradation types. When the detector which is trained with the aim of detection of UN examples with severity 0.1 is used to detect the IFGSM perturbations, it yields 82.86% and 87.23% accuracy on the CIFAR-10 and CIFAR-100 datasets, respectively. Moreover, the confidence in adversarial perturbation detection increases significantly when the PGD attack which is considered one of the strongest first-order universal adversaries is brought for detection. On both datasets, the UN-corrupted trained detector yields almost perfect PGD perturbation detection accuracy. The degradation induced by UN corruption with severity 0.1 is found quasi-imperceptible (the third row of UN corrupted examples in Fig. 4) similar to the adversarial perturbations. Results of this interesting study are reported in Table 5.

Table 7: Common corruption mitigation on the CIFAR-10 dataset using ResNet and Wide-ResNet16-8. In brief, the noise remover trained with severity coming at the test time shows better robustness as compared to training on different severity levels. However, the unseen severity training can boost the performance multiple folds. The detailed interpretative caption is given in the text as well.

Data	Mitigation Severity																	
	GN			UN			SPN			SN			IN			SPKN		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
ResNet: 91.81% clean accuracy																		
1N	32.97			13.84			20.48			32.73			51.73			44.37		
1P	75.97	71.92	64.04	28.24	31.22	40.72	83.62	79.05	73.67	76.62	70.06	59.82	82.9	83.28	78.1	78.95	76	68.98
2N	19.46			15.4			13.14			18.13			30.82			32.51		
2P	71.69	72.13	64.68	42.45	48.15	60.0	81.4	78.79	73.33	71.08	69.72	62.02	82.9	82.95	78.35	77.0	75.92	68.62
3N	15.22			18.19			10.96			14.97			21.52			18.6		
3P	56.27	63.68	63.34	58.12	65.24	69.47	64.79	75.03	71.92	57.12	63.04	62.75	82.46	83.14	77.93	61.0	66.53	68.55
Wide-ResNet16-8: 93.41% clean accuracy																		
1N	45.27			15.97			20.2			45.33			63.52			58.17		
1P	80.05	75.81	66.76	30.83	34.42	42.94	87.07	82.45	77.01	80.44	73.97	62.37	86.25	86.34	82.37	82.54	80.15	72.34
2N	25.06			16.55			11.09			25.72			35.79			46.02		
2P	75.74	75.8	67.36	45.45	51.96	62.97	85.27	82.06	76.85	75.15	74.23	65.32	86.33	86.43	82.6	80.59	79.4	72.7
3N	15.16			21.72			10.32			17.77			21.74			26.0		
3P	60.84	67.61	67.76	61.28	69.88	73.67	69.99	79.14	75.26	61.8	67.51	66.03	86.21	86.01	81.84	64.81	70.26	71.79

In contrast to earlier cross-degradation detection findings, detectors trained on adversarial perturbations are found significantly generalized in detecting common corruptions even across different severity levels. It can be understood from the fact that adversarial perturbations are significantly subtle and hence if the detector can identify such perturbation, it can better detect common corruption having a similar or higher amount of perturbation. On the CIFAR-10 dataset when the training degradation is IFGSM and testing is done on common corruptions, at least 87.02% detection accuracy is observed. On each dataset, IN corruption with the lowest severity is found highly challenging to detect. The high detection robustness of adversarial perturbation against common corruption is shown in Table 6.

3.2 Mitigation

The second objective of this research is to be able to mitigate the impact of these highly adverse degradations to preserve recognition performance. Tables 7 and 8 show the mitigation results on the CIFAR-10 and CIFAR-100 datasets, respectively. *For both these tables, the numbers in row iN represent the performance of the CNN when under the effect of the common corruption generated with i^{th} severity level. The row with iP contains the mitigation performance of the ith severity corruption using the proposed algorithm. For instance, GN-1N: accuracy on samples corrupted with severity-1 Gaussian noise. GN-1P-1 represents the mitigation accuracy on severity-1 Gaussian noise corrupted examples when the proposed noise remover is trained using severity-1 examples.* In numerical terms: GN-1N can reduce the accuracy of Wide-ResNet28-10 classifier 24.71%, which increases to 51.95% (GN-1P-1), 47.31% (GN-1P-2), 36.85% (GN-1P-3) when the proposed noise remover is trained using the Gaussian noise images with severity 1, 2, and 3, respectively.

3.2.1 Results on CIFAR Datasets

The ResNet (RN) and Wide-ResNet16-8 (WRN) classifiers yield 91.81% and 93.41% classification accuracy on the clean test images of the CIFAR-10 dataset, respectively. The RN classifier can be seen as a deep classifier whereas, the WRN classifier is both deep and wide. We see that every corruption even added with the lowest severity can reduce the performance of both classifiers. In terms of corruption, among all the noises, SPN corruption was found highly stealthy.

To extensively study the impact of noise removal, we have performed the experiments under both the same and different severity levels. For instance, when the proposed noise remover is trained using severity-1 of Gaussian noise, it can increase the performance to 75.97%, 71.69%, and 56.27% when the corrupted images with severity one (32.97%), two (19.46%), and three (15.22%) used for classification, respectively. This showcases that the proposed algorithm can mitigate corruption regardless of severity differences. In most

Table 8: Common corruption mitigation on the CIFAR-100 dataset using Wide-ResNet28-10 and Wide-ResNet16-8. Common corruption mitigation on the CIFAR-10 dataset using ResNet and Wide-ResNet16-8. In brief, the noise remover trained with severity coming at the test time shows better robustness as compared to training on different severity levels. However, the unseen severity training can boost the performance multiple folds. The detailed interpretative caption is given in the text as well.

Data	Mitigation Severity																	
	GN			UN			SPN			SN			IN			SPKN		
	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3	1	2	3
Wide-ResNet28-10: 76.21% clean accuracy																		
1N	24.71			2.51			6.64			22.95			26.81			31.6		
1P	51.95	47.31	36.85	17.68	12.62	9.22	60.92	54.52	45.39	50.89	45.32	36.11	61.82	61.43	60.41	54.7	52.1	41.71
2N	12.5			4.35			10.88			10.87			10.79			20.82		
2P	46.6	46.41	37.74	34.62	24.54	18.44	61.0	54.72	45.05	43.71	45.0	36.68	61.64	61.34	60.1	50.99	51.55	42.62
3N	6.57			9.44			1.00			6.67			6.99			8.18		
3P	31.55	36.75	37.35	43.78	40.07	31.23	39.13	48.46	45.99	29.51	35.52	37.44	61.1	61.1	60.25	31.13	39.64	42.56
Wide-ResNet16-8: 74.57% clean accuracy																		
1N	20.09			2.26			4.12			17.44			21.59			25.68		
1P	53.17	48.46	38.55	18.75	12.68	8.49	61.18	55.69	46.5	52.32	46.2	37.42	62.38	61.5	60.77	55.86	53.29	43.27
2N	9.17			3.37			8.02			7.78			7.97			15.4		
2P	48.23	47.88	39.31	35.87	25.06	18.24	61.06	55.58	46.07	45.5	46.57	38.48	62.19	61.55	60.92	52.15	52.53	44.02
3N	4.03			6.7			1.06			3.88			4.66			6.27		
3P	33.44	38.11	39.37	45.35	42.12	32.13	41.27	49.76	47.18	30.12	38.18	38.87	61.7	61.47	60.7	31.71	40.91	43.42

cases of severe common corruptions, the performance boost is as high as 6 times (10.96% to 64.79 in case of RN) to 7 times (10.32% to 69.99% in case of WRN).

In contrast to CIFAR-10, the impact of noise corruption is significantly higher on the CIFAR-100 dataset, which might be due to a large number of classes leading to higher complexity. SPN corruption is again able to reduce the performance to the lowest accuracy, i.e., 1.0% which is random chance accuracy for 100 classes. However, again as observed on the CIFAR-10 dataset, the proposed noise remover algorithm can boost the recognition performance substantially. Similar to the CIFAR-10 dataset, the experiments are extensively performed under both the same and different corruption severity. The experiments are performed using two wider and deeper architectures namely Wide-ResNet28-10 (WRN28-10) and Wide-ResNet16-8 (WRN16-8). The analysis can be broken down broadly into two pieces: (i) degradation impact on the performance of both classifiers under different common corruption and (ii) improvement in the recognition accuracy when the images are purified using the proposed noise remover algorithm.

The WRN28-10 and WRN16-8 yield an accuracy of 76.21% and 74.57% on the clean images of the CIFAR-100 dataset, respectively. However, the performance suffers hugely when the distribution of the testing images changes due to different corruptions. As observed earlier, the WRN16-8 architecture being shallow as compared to WRN28-10 is found less robust in handling corruption. For instance, the performance of the WRN28-10 classifier drops to 22.95%, and the performance of WRN16-8 drops further to 17.44% when the shot noise (SN) with severity one is used to attack the classifiers. The impact of other common corruption also gets noticed significantly on both the classifiers and hence we reaffirm that no corruption should be ignored and the severity of the corruption is also an important factor to consider.

The proposed algorithm can increase recognition performance on the CIFAR-100 dataset as well reflecting that it is dataset agnostic. Similar to the improvement on CIFAR-10, the increase in the CIFAR-100 is also manifold. For instance, on the IN corruption with severity three, the proposed algorithm boosts the recognition performance of WRN28-10 from 6.99% to 61.1%. The boost is even higher when the shallow classifier (WRN16-8) is used, i.e., the performance is boosted from 4.66% to 61.7%. The performance improves by more than 39 and 40 times in cases where perturbations can reduce the performance to a random chance level (i.e., 1.0%). We want to highlight that the proposed noise remover does not utilize any knowledge of the classifiers and hence can be effectively plugged with 'any' machine learning classifiers.

Table 9: Common corruption mitigation results on the Tiny ImageNet dataset. N (↓) and P (↑) represent the performance on noisy and mitigated images, respectively. The proposed noise remover is trained on an unseen severity level (i.e., severity 1 which is denoted as -1). Similar to the other datasets, the proposed noise remover can restore the recognition performance of any classifier operating on any large-scale dataset. A few values are highlighted to quickly highlight the strength of the proposed algorithm.

Train ↓	DenseNet				MobileNet				VGG16			
Test →	severity 2		severity 3		severity 2		severity 3		severity 2		severity 3	
	N (↓)	P (↑)	N (↓)	P (↑)	N (↓)	P (↑)	N (↓)	P (↑)	N (↓)	P (↑)	N (↓)	P (↑)
GN-1	22.43	39.16	9.33	33.89	20.60	33.37	8.18	27.06	29.23	37.43	16.19	31.95
UN-1	5.39	23.11	13.93	33.61	4.45	17.68	11.95	27.05	9.27	21.87	20.99	30.96
SPN-1	**0.81**	**39.50**	0.52	16.60	**0.85**	**33.73**	0.57	13.45	**1.46**	**36.97**	0.68	17.45
SN-1	20.04	38.61	9.46	33.78	19.35	32.45	8.79	27.96	27.79	35.97	16.61	32.84
IN-1	14.67	43.99	**8.20**	**43.90**	16.69	38.26	**8.87**	**38.00**	22.80	39.47	**15.06**	**39.12**
SPKN-1	34.14	40.55	13.55	34.97	31.38	35.06	13.40	30.15	36.39	36.59	21.13	33.77

3.2.2 Results on Tiny-ImageNet

We have also demonstrated the strength of the proposed noise remover network by showing that it can scale to a large-scale dataset both in terms of classes and resolution of the images. For this purpose, we have used the **Tiny Imagenet** dataset Le & Yang (2015) which contains high-resolution images of size $64 \times 64 \times 3$ of 200 object classes. Not only the dataset, but we have extended the CNN architectures as well to establish the classifier-agnostic nature of the proposed generative robustness network architecture. We have used *DenseNet-121* Huang et al. (2017), *MobileNet* Howard et al. (2017), and *VGG-16* Simonyan & Zisserman (2014) pre-trained on the ImageNet and fine-tuned for object classification on Tiny ImageNet training images. DenseNet, MobileNet, and VGG architectures yield 62.08%, 57.53%, and 50.41% classification accuracy on the clean test images of the dataset, respectively. We have evaluated the performance of the proposed noise remover against the unseen severity level of the corruption to check the generalizability. In other words, severity level one is used to train the noise remover network and the testing has been performed on the corrupted images using severity levels two and three.

The Salt & Pepper noise (SPN) common corruption is proven most severe and reduces the performance of each network by approximately close to 0%. However, the proposed noise remover network helps in restoring the classification performance of each network by a significant margin. For instance, the severity level of two corrupted SPN images reduces the performance of DenseNet from 62.08% to **0.81**%, which after mitigation using the proposed framework is restored to **39.50**%. Similarly, the vulnerable performance of the VGG and MobileNet architecture impacted by SPN-2, increases by **35.51**% and **32.88**%, respectively. The robustness is not only observed in SPN but also against each other corruption and by multiple factors. These experiments further support our assertion that the proposed generative robustness network can be *applied against any dataset, corruption, classifier whether seen at the times of training or not.* The results of the proposed robustness on the Tiny ImageNet dataset are reported in Table 9.

3.2.3 Results on ImageNet

To further expand the evaluation of the impact of the proposed generative robustness network, we have now utilized the high-resolution subset[1] of the *'ImageNet'* dataset Deng et al. (2009). Similar to the experiments on Tiny ImageNet, we have used three classical CNN architectures. DenseNet, MobileNet, and VGG architecture yield 96.10%, 95.20%, and 92.50% classification accuracy on the clean test images of the subset, respectively. The generalization of the proposed framework is evaluated by training on severity level one corruption and tested on levels two and three. The results of the proposed algorithm on the ImageNet subset are reported in Table 10. Several interesting results are observed in the high-resolution dataset: (i) the proposed noise remover is agnostic to image resolution and can effectively mitigate the impact of corruption by a huge margin and (ii) surprisingly, the noise remover also helps in achieving better performance than

[1] https://github.com/fastai/imagenette

Table 10: Common corruption mitigation results on the subset of **'ImageNet'** dataset Deng et al. (2009). N (↓) and P (↑) represent the performance on noisy and mitigated images, respectively. The proposed noise remover is trained on an unseen severity level (i.e., severity 1 which is denoted as -1). Similar to multiple datasets, the proposed noise remover can restore the recognition performance of any classification operating on any large-scale dataset. A few values are highlighted to quickly highlight the strength of the proposed algorithm.

Train ↓	DenseNet				MobileNet				VGG16			
Test →	severity 2		severity 3		severity 2		severity 3		severity 2		severity 3	
	N (↓)	P (↑)	N (↓)	P (↑)	N (↓)	P (↑)	N (↓)	P (↑)	N (↓)	P (↑)	N (↓)	P (↑)
GN-1	75.40	95.10	55.10	88.80	67.80	92.90	**42.60**	**88.80**	65.60	92.00	35.10	84.00
UN-1	27.10	80.80	62.40	87.30	29.70	77.70	52.80	88.30	**12.40**	**76.10**	44.60	85.20
SPN-1	**7.90**	**91.00**	**1.00**	**72.70**	**20.50**	**91.40**	21.80	70.40	**0.00**	**90.90**	**0.00**	**56.80**
SN-1	73.20	95.00	51.50	93.10	62.30	90.60	41.70	87.20	71.20	91.00	44.20	87.90
IN-1	61.40	**97.00**	48.20	**96.80**	55.80	93.90	38.30	92.80	50.60	92.20	**27.20**	**92.40**
SPKN-1	86.50	96.30	61.50	94.30	81.70	93.50	56.00	90.60	84.50	92.10	63.00	89.30

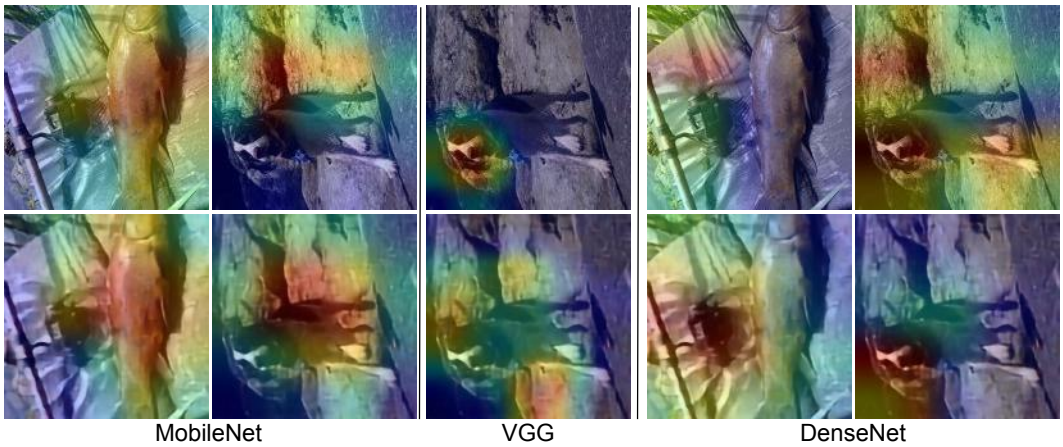

Figure 5: Grad-CAM Selvaraju et al. (2017) visualization on the clean and purified images shows that the proposed framework can restore the focus of the CNNs to the critical regions. The first row shows the visualizations of the clean images and the last row shows the purified images.

clean images in some cases. For instance, in case 1, the SPN corruption reduces the performance of VGG to **0.00**%, which the proposed noise remover restored to **90.90**% and **56.80**% for unseen severity levels two and three, respectively. In the instance related to case 2, the DenseNet classifier yields 96.10% accuracy on the clean test of the dataset which got improved to 97.00% when the IN corruption with severity two purified using the proposed noise remover trained on IN with severity one.

As shown in Fig. 5, we observe that the proposed framework can ensure that the different classification networks focus on a more classification-relevant localized region in the purified images which sometimes is even better than the focus of the CNNs on the real/clean images. It might be the result of enhanced recognition performance which otherwise suffers due to the corruption of images and their corresponding focus regions being taken into account by the networks.

3.2.4 Mitigation of Unseen Adversarial Perturbations

In this section, we study the strength of the proposed noise remover algorithm when trained to remove the common corruption but tested on the adversarial perturbations. The extensive evaluation has been performed using two adversarial perturbation algorithms used above namely PGD and IFGSM. The PGD perturbations are generated for 40 epochs with standard strength $\epsilon = 0.03$ with a random restart value of

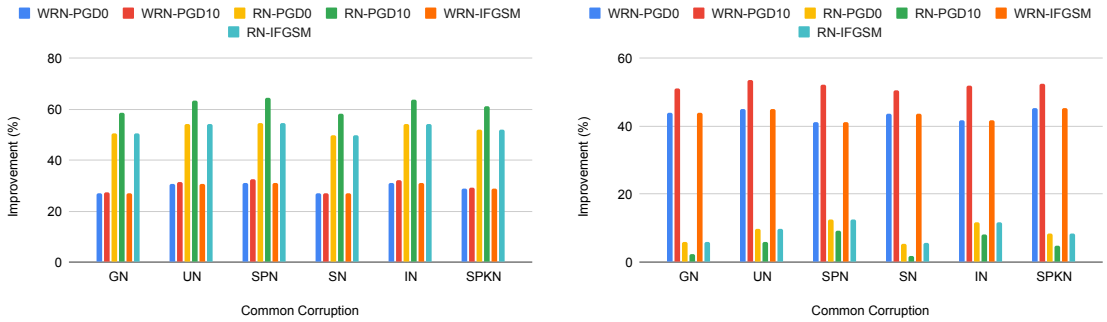

Figure 6: Improvement in recognition performance on adversarial examples on CIFAR-10 where the proposed noise remover is trained on common corruptions. The left figure shows the performance on the RN classifier and the right bars show the adversarial mitigation on the WRN16-8 classifier.

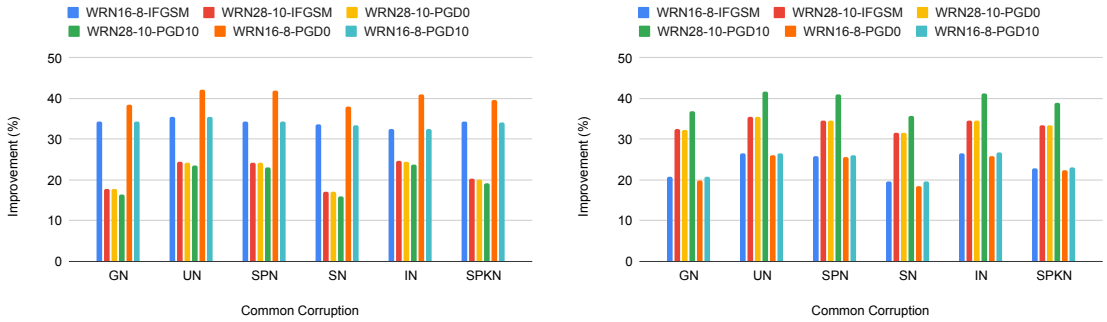

Figure 7: Improvement in recognition performance on adversarial examples on CIFAR-100 where the proposed noise remover is trained on common corruptions. The left figure shows the performance on WRN16-8 and the right bars show the adversarial mitigation on the WRN28-10 classifier.

0 (PGD0) and 10 (PGD10). The IFGSM perturbation is generated using 40 epochs with strength $\epsilon = 0.03$. Both the perturbations are generated using ResNet and Wide-ResNet16-8 models on the CIFAR-10 dataset and WRN16-8 and WRN28-10 on the CIFAR-100 dataset.

When the classifiers are attacked using the adversarial perturbations, the proposed noise remover which is trained to remove the common corruption shows significant robustness and improves the performance on the adversarial examples. When the ResNet classifier is protected using the GN corrupted trained noise remover, it shows an improvement of 27.41% and 58.7% when the PGD adversarial images are generated from WRN and RN are used, respectively. The lower improvement in the WRN adversarial examples can be attributed to the fact that the PGD adversarial examples are transferred from WRN to RN and it shows lower attack success. In terms of corruption, the SPN-trained noise remover shows the best average performance improvement of 42.87%. UN and IN corruption-trained noise removers also performed similarly, behind by 0.4%.

As mentioned above when the adversarial examples are transferred from one CNN to another, they yield lower attack success; and leave less space for improvement. However, when the perturbation is generated from the same network, the attack success is much higher; conversely, the improvement in adversarial mitigation is also higher. For instance, when the PGD0 generated from the WRN is used to attack the WRN classifier, the proposed noise remover shows 44.11% improvement as compared to 5.78% improvement when RN-generated noise is used to attack the classifier. The transference of the adversarial perturbation is an interesting concern where several research works are recently trying to improve the success of the attack against unseen classifiers Gao et al. (2021); Inkawhich et al. (2019); Li et al. (2020). It is to be noted that the proposed framework does not utilize any knowledge of the perturbation or the classifier and can handle even the highly successful transferable perturbations. The improvement is observed on both the image classifiers and datasets, therefore

Table 11: Mitigation results on the CIFAR-10 dataset. The proposed 'GRN' outperforms traditional denoising (DCAE) and existing algorithms by a significant margin across each corruption. Raw represents the accuracy of the corrupted test set.

Noise	**CNN**	**Raw**	**Salman et al. (2020)**	**Xie et al. (2019)**	**DCAE**	**GRN**
GN	ResNet	32.97	55.65	57.21	54.21	**75.97**
	WRN	45.27	60.04	45.69	59.30	**80.05**
IN	ResNet	51.73	58.17	65.20	59.75	**82.90**
	WRN	63.52	62.34	58.36	53.87	**86.25**
SN	ResNet	32.73	64.53	48.31	56.47	**76.62**
	WRN	45.33	58.76	46.89	60.02	**80.44**
SPKN	ResNet	44.37	55.94	71.05	57.90	**78.95**
	WRN	58.17	69.93	57.01	61.82	**82.54**
UN	ResNet	18.19	64.24	66.53	53.99	**69.47**
	WRN	21.72	67.93	61.00	58.08	**73.67**
SPN	ResNet	20.48	56.55	42.18	50.07	**83.62**
	WRN	20.20	60.98	35.76	54.02	**87.07**

showing the efficacy of the proposed mitigation algorithm across degradations, datasets, and classifiers. The results on CIFAR-10 and CIFAR-100 datasets are reported in Fig.(s) 6 and 7, respectively.

3.3 Ablation Studies

The ablation studies are conducted concerning several important components used in the proposed research: (i) varying degradation detection network, (ii) utilization of denoising convolutional autoencoder (DCAE) having a single output, (iii) degradation detection without noise augmentation, and (iv) can we handle grayscale images?

Degradation Detection Network: In the proposed research, we have used the Wide-ResNet10-2 (WRN10-2) Zagoruyko & Komodakis (2016) for the classification of degradation examples. We have further performed an ablation study using another popular classifier namely VGG Simonyan & Zisserman (2014). We have observed across the set of experiments, VGG yields lower performance as compared to WRN10-2. For instance, when the GN corrupted trained detector is tested on the UN-1, the VGG architecture yields 7% lower on the CIFAR-100 dataset as compared to WRN10-2. The performance difference even increased to 22% when the evaluation was performed on the CIFAR-10 dataset.

Impact of Augmenting Noise: When in the proposed framework, the noise removed from the images is not augmented with the input images; the performance of the proposed degradation detection network reduces 3%-6%. It shows the advantage of the auxiliary noise information obtained from the proposed noise remover network in the proposed detection network as compared to the networks utilizing only RGB images.

DCAE vs. Proposed: The mitigation performance of the proposed noise remover network with two branches (one as clean image and another as noise pattern) yields at least 6% better performance than the traditional single branch (mapping noisy images to clean images) denoising convolutional autoencoder. The probable reason might be that these traditional denoising systems map each kind of noisy image into ideal clean images without giving importance to the variation present in different noise signals.

Performance on Gray-scale Images: To evaluate the performance of the proposed noise mitigation framework, we consider the F-MNIST dataset Xiao et al. (2017) and build a custom CNN model with 5 layers. The model yields 91.45% accuracy on the clean test images. The proposed denoiser is able the boost the recognition performance dropped under each corruption. For instance, on GN, UN, and SPN, the performance boost observed is at least 15%, 50%, and 48%, respectively. The above analysis shows that the proposed algorithm can handle the (i) custom models as well not just the wide and deep networks and (ii) can boost the recognition performance on grayscale images as well.

Table 12: Comparison results against SOTA works. "Clean" indicates Top-1 clean accuracy (%) (higher is better). "mCE" shows the performance (%) over 15 corruption types (less is better).

Algorithm	CIFAR10-C		CIFAR100-C	
	Clean↑	mCE↓	Clean↑	mCE↓
Kim et al. Kim et al. (2021)	75.3	45.6	57.9	48.1
Yang et al. Yang et al. (2022)	77.2	41.9	58.0	46.4
Zou et al. Zou et al. (2020)	62.3	45.1	55.2	45.8
Hendrycks et al. Hendrycks et al. (2019)	79.5	43.4	60.6	44.9
DuTaNet	**81.7**	**39.6**	**62.4**	**42.2**

Comparison with existing state-of-the-art works: Extensive experiments performed using multiple datasets and CNN architectures showcase the importance of the proposed noise remover and detection algorithm in building unified security and robustness. To further demonstrate the effectiveness of the proposed work, a comprehensive comparison with existing state-of-the-art (SOTA) algorithms Salman et al. (2020); Xie et al. (2019) has also been performed. The comparison has been performed using multiple corruption algorithms that not only reflect the capacity (in increasing the robustness against individual corruptions) but also the generalizability (handling unseen corruptions). As seen each corruption can degrade the recognition performance of the image classification networks. The proposed and existing mitigation algorithms aim to restore recognition accuracy. The results reported in Table 11 show that the existing algorithms and traditional image-denoising architectures miserably failed in doing so. Further, these algorithms are not found generalized in handling corruption and yield poor recognition accuracy. On top of that, the existing algorithms have heavy computational demand to develop an image enhancement algorithm. For instance, Xie et al. Xie et al. (2019) have utilized the concept of adversarial training and the algorithm took 52 hours on 128 V100 GPUs to defend the PGD attack only.

We have also performed an extensive comparison with recent SOTA image-denoising algorithms and the results are reported in Table 12. The performance of the proposed algorithm is compared using metrics such as performance on clean images (higher the better) and mCE (lower the better), following the protocols of the existing works. Kim et al. Kim et al. (2021) have used the invertible encoder-decoder architecture, where the first stage learns the object recognition network and the second stage produces low-quality and high-quality (HQ) images. Due to the high dependency on high-quality images, the algorithm is found less resilient across multiple forms of corruption. Yang et al. Yang et al. (2022) proposed the feature distillation method to produce HQ features. As can be seen from the results, the proposed algorithm outperforms the existing algorithms by a significant margin and is found generalized in handling unseen noise corruptions. Further, apart from these algorithms, we have compared our algorithm Byun et al. Byun et al. (2021) and IDR Zhang et al. (2022) in terms of the PSNR metric. Byun et al. Byun et al. (2021) propose the network to model the Gaussian-Poisson distribution. Along with the requirement of a Gaussian-Poisson pair, the algorithm is computationally heavy having $2,512$ MB as compared to the proposed algorithm. The PSNR value of the proposed algorithm is at least 1.5% higher than Byun et al. (2021) on the CIFAR-10 and CIFAR-100 datasets. In comparison to IDR Zhang et al. (2022) which works on iterative generation of noisier-noisy images, the proposed algorithm yields 3.2% and 3.5% higher PSNR on CIFAR-10 and CIFAR-100, respectively.

3.4 Broader Impact

The broader impact of the proposed research can be effectively evaluated across three dimensions: (i) classifier, (ii) dataset, and (iii) degradation. These dimensions can be further broken down to reflect the universality of the proposed defense mechanism. For instance, the classifiers can be broken into (i) shallow, (ii) deep, and (iii) wide architectures. In this research, we have evaluated the robustness of each category of classifier/CNN. The custom networks used for F-MNIST are shallow, VGG and DenseNet can be grouped into deep architectures, and Wide-ResNet is wide architectures. The effectiveness of the framework with a variety of classifiers shows that the proposed algorithm can be utilized with any classifier to defend against the impact of corruption.

For each dataset, we have evaluated the performance of the proposed algorithm again in multiple aspects such as classes (we have used datasets containing 10, 100, and 200 classes), channels (RGB vs. grayscale), and resolution (low-resolution CIFAR-10/CIFAR-100 and high-resolution ImageNet). The last broad dimension we consider is the development of a defense algorithm against a variety of degradations including common corruptions and adversarial perturbations. These degradations cover the different types of corruption and their severity levels are also varied to address a critical concern for universal resiliency. Our key assertion is that the proposed defense is generalized, which we have thoroughly evaluated in terms of unseen severity levels and degradation-type training-testing settings. All in all, the proposed research is a big step towards building a universal defense system to protect the robustness and integrity of deep machine learning.

4 Conclusion

Image corruption, whether common or deliberate, can adversely impact the performance of neural network-based classification systems and reduce their real-world applicability. While existing literature has addressed these problems individually, on top of that the attempts made so far yield poor generalizability in handling unseen corruptions Geirhos et al. (2018); Lopes et al. (2019). We discussed that common corruptions and adversarial perturbation are similar and should be handled in conjunction to create truly resilient algorithms. With this objective, we presented the unified corruption detection and mitigation framework termed a generative robustness network. Results on multiple publicly available databases demonstrate its efficacy in both detecting the presence of corruptions in input images as well as mitigating their adverse effect to maintain classification performance levels. We also establish a possible connection between common and deliberate corruptions empirically as we show that the network can scale well across types of corruptions, successfully performing on deliberate corruptions while being trained only on common corruptions. Future research in this area can focus on further enhancing the high-level blocks of the proposed architecture and leveraging it for a wide variety of real-world use cases.

References

Ahmed Abusnaina, Yuhang Wu, Sunpreet Arora, Yizhen Wang, Fei Wang, Hao Yang, and David Mohaisen. Adversarial example detection using latent neighborhood graph. In *IEEE/CVF International Conference on Computer Vision*, pp. 7687–7696, 2021.

Akshay Agarwal, Mayank Vatsa, Richa Singh, and Nalini K. Ratha. Noise is inside me! generating adversarial perturbations with noise derived from natural filters. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, June 2020.

Akshay Agarwal, Gaurav Goswami, Mayank Vatsa, Richa Singh, and Nalini K. Ratha. Damad: Database, attack, and model agnostic adversarial perturbation detector. *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–13, 2021a. doi: 10.1109/TNNLS.2021.3051529.

Akshay Agarwal, Richa Singh, Mayank Vatsa, and Nalini Ratha. Image transformation-based defense against adversarial perturbation on deep learning models. *IEEE Transactions on Dependable and Secure Computing*, 18(5): 2106–2121, 2021b. doi: 10.1109/TDSC.2020.3027183.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. *Advances in Neural Information Processing Systems*, 33:16048–16059, 2020.

Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. Deep learning through the lens of example difficulty. *Advances in Neural Information Processing Systems*, 34, 2021.

Jaeseok Byun, Sungmin Cha, and Taesup Moon. FBI-denoiser: Fast blind image denoiser for poisson-gaussian noise. In *IEEE/CVF CVPR*, pp. 5768–5777, 2021.

Dan A Calian, Florian Stimberg, Olivia Wiles, Sylvestre-Alvise Rebuffi, Andras Gyorgy, Timothy Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. *arXiv preprint arXiv:2104.01086*, 2021.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Zhijie Deng, Xiao Yang, Shizhen Xu, Hang Su, and Jun Zhu. Libre: A practical bayesian approach to adversarial detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 972–982, 2021.

Nathan Drenkow, Neil Fendley, and Philippe Burlina. Attack agnostic detection of adversarial examples via random subspace analysis. In *IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 472–482, 2022.

Lianli Gao, Zijie Huang, Jingkuan Song, Yang Yang, and Heng Tao Shen. Push & pull: Transferable adversarial examples with attentive attack. *IEEE Transactions on Multimedia*, 2021.

Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. *Advances in neural information processing systems*, 31, 2018.

Gaurav Goswami, Akshay Agarwal, Nalini Ratha, Richa Singh, and Mayank Vatsa. Detecting and mitigating adversarial perturbations for robust face recognition. *International Journal of Computer Vision*, 127(6):719–742, 2019.

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*, pp. 770–778, 2016.

Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 15262–15271, 2021.

Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4700–4708, 2017.

Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7066–7074, 2019.

Ziheng Jiang, Chiyuan Zhang, Kunal Talwar, and Michael C Mozer. Characterizing structural regularities of labeled data in overparameterized models. In *International Conference on Machine Learning*, pp. 5034–5044. PMLR, 2021.

Hoki Kim, Woojin Lee, and Jaewook Lee. Understanding catastrophic overfitting in single-step adversarial training. *arXiv preprint arXiv:2010.01799*, 2020.

Insoo Kim, Seungju Han, Ji-won Baek, Seong-Jin Park, Jae-Joon Han, and Jinwoo Shin. Quality-agnostic image recognition via invertible decoder. In *IEEE/CVF CVPR*, pp. 12257–12266, 2021.

Ezgi Korkmaz. Adversarial training blocks generalization in neural policies. In *NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications*, 2021.

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

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pp. 99–112. Chapman and Hall/CRC, 2018.

Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning transferable adversarial examples via ghost networks. In *AAAI Conference on Artificial Intelligence*, volume 34, pp. 11458–11465, 2020.

Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens Van Der Maaten. Exploring the limits of weakly supervised pretraining. In *European Conference on Computer Vision*, pp. 181–196, 2018.

Devraj Mandal, Sanath Narayan, Sai Kumar Dwivedi, Vikram Gupta, Shuaib Ahmed, Fahad Shahbaz Khan, and Ling Shao. Out-of-distribution detection for generalized zero-shot action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9985–9993, 2019.

Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL `https://proceedings.neurips.cc/paper/2016/file/0ed9422357395a0d4879191c66f4faa2-Paper.pdf`.

Eric Mintun, Alexander Kirillov, and Saining Xie. On interaction between augmentations and corruptions in natural corruption robustness. *Advances in Neural Information Processing Systems*, 34, 2021.

Anibal Pedraza, Oscar Deniz, and Gloria Bueno. Really natural adversarial examples. *International Journal of Machine Learning and Cybernetics*, pp. 1–13, 2021.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

Jie Ren, Peter J Liu, Emily Fertig, Jasper Snoek, Ryan Poplin, Mark Depristo, Joshua Dillon, and Balaji Lakshminarayanan. Likelihood ratios for out-of-distribution detection. *Advances in Neural Information Processing Systems*, 32, 2019.

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*, pp. 53–69. Springer, 2020.

Hadi Salman, Mingjie Sun, Greg Yang, Ashish Kapoor, and J Zico Kolter. Denoised smoothing: A provable defense for pretrained classifiers. *Advances in Neural Information Processing Systems*, 33:21945–21957, 2020.

Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

Alex Serban, Erik Poll, and Joost Visser. Adversarial examples on object recognition: A comprehensive survey. *ACM Computing Surveys (CSUR)*, 53(3):1–38, 2020.

Ali Shafahi, Mahyar Najibi, Zheng Xu, John Dickerson, Larry S Davis, and Tom Goldstein. Universal adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5636–5643, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Apoorv Vyas, Nataraj Jammalamadaka, Xia Zhu, Dipankar Das, Bharat Kaul, and Theodore L Willke. Out-of-distribution detection using an ensemble of self supervised leave-out classifiers. In *European Conference on Computer Vision*, pp. 550–564, 2018.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 501–509, 2019.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10687–10698, 2020.

Zhou Yang, Weisheng Dong, Xin Li, Jinjian Wu, Leida Li, and Guangming Shi. Self-feature distillation with uncertainty modeling for degraded image recognition. In *ECCV*, pp. 552–569. Springer, 2022.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Huan Zhang, Hongge Chen, Zhao Song, Duane Boning, Inderjit S Dhillon, and Cho-Jui Hsieh. The limitations of adversarial training and the blind-spot attack. *arXiv preprint arXiv:1901.04684*, 2019.

Jiliang Zhang and Chen Li. Adversarial examples: Opportunities and challenges. *IEEE transactions on neural networks and learning systems*, 31(7):2578–2593, 2019.

Yi Zhang, Dasong Li, Ka Lung Law, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. IDR: Self-supervised image denoising via iterative data refinement. In *IEEE/CVF CVPR*, pp. 2098–2107, 2022.

Zhengxia Zou, Sen Lei, Tianyang Shi, Zhenwei Shi, and Jieping Ye. Deep adversarial decomposition: A unified framework for separating superimposed images. In *IEEE/CVF CVPR*, pp. 12806–12816, 2020.