Are Adversarial Robustness and Common Perturbation Robustness Independant Attributes ?

Alfred LAUGROS[1,2], Alice CAPLIER[1], and Matthieu OSPICI[2]

[1]Univ. Grenoble Alpes
[2]Atos
{*alfred.laugros,alice.caplier*}*@grenoble-inp.fr*
matthieu.ospici@atos.net

Abstract

Neural Networks have been shown to be sensitive to common perturbations such as blur, Gaussian noise, rotations, etc. They are also vulnerable to some artificial malicious corruptions called adversarial examples. The adversarial examples study has recently become very popular and it sometimes even reduces the term "adversarial robustness" to the term "robustness". Yet, we do not know to what extent the adversarial robustness is related to the global robustness. Similarly, we do not know if a robustness to various common perturbations such as translations or contrast losses for instance, could help with adversarial corruptions. We intend to study the links between the robustnesses of neural networks to both perturbations. With our experiments, we provide one of the first benchmark designed to estimate the robustness of neural networks to common perturbations. We show that increasing the robustness to carefully selected common perturbations, can make neural networks more robust to unseen common perturbations. We also prove that adversarial robustness and robustness to common perturbations are independent. Our results make us believe that neural network robustness should be addressed in a broader sense.

1. Introduction

Deep Neural Networks have been shown to be very efficient in image processing tasks such as content classification [15], face recognition [31] or object detection [24]. Despite their good performances on academic datasets, artificial neural networks are vulnerable to common perturbations like blur, lightning variations or colorimetry changes [14, 3]. These perturbations are often encountered in industrial applications and can make some models useless.

Some techniques can be used to increase neural network robustness to such perturbations. Data augmentation approaches [28] or fine tuning techniques [36] are broadly used to protect neural networks. Regularization techniques in general are useful to build models robust to traditional perturbations [35]. Despite recent great advances, neural networks are not successful enough at dealing with corrupted images coming from real world applications [4].

Deep neural networks are also vulnerable to slightly modified samples called adversarial examples [30]. They consist in an addition of malicious patterns that can completely disturb the behavior of a neural network.

A lot of defense strategies have been proposed to make neural networks more robust to adversarial examples. Distillation learning can be used to make neural networks more stable [23]. Introducing adversarial examples in the training procedure can decrease neural networks sensitivity to these attacks [22]. Additional modules such as autoencoder [9], or GAN [26], have been used to protect neural networks from adversarial corruptions. Regularization is also a standard procedure to make neural networks more robust to adversarial samples [25]. However, none of these techniques succeeds in making a neural network perfectly invariant to adversarial examples.

In some recent studies, the expression "noise robustness" or the expression "adversarial robustness" are reduced to "robustness" only [2, 20]. However, we do not know to what extent these fields are linked. Some recent works show that the salient points used by adversarially trained models to understand images are close to the ones used by humans [33]. Then, one could expect these models to be robust to the same kinds of perturbations as humans. We wonder if making neural networks more robust to adversarial perturbations could also make them more robust to common perturbations and vice versa. To better understand neural net-

work global robustness, we want to study the hypothetical correlations between different kinds of robustnesses.

As part of our studies, we provide a thorough method to build a set of common perturbations in order to estimate the robustness of neural networks. We carry out experiments about making neural networks robust to unseen common perturbations. Finally, we study eventual correlations between the adversarial robustness and the robustness to common perturbations.

2. Background and Related Works

2.1. Common perturbations

A few academic datasets are used to compare state of the art networks: [15, 21, 18]. Academic datasets are really useful to researchers, but they do not necessarily cover the various perturbations encountered in real application cases. In the CelebA dataset for instance, lightning conditions, face positions and colorimetry are constant [21]. But in real application cases, various transformations can be introduced by sensor characteristics, lighting conditions or motions. Image processing or data transmission can also introduce unexpected distortions. We call these distortions common perturbations. We consider that common perturbations are transformations that are often encountered in industrial contexts but which are generally absent from academic datasets. This definition includes traditional additive noises such as Gaussian or salt-pepper noises. It also covers global changes in lightning conditions, contrast or colorimetry. Geometric transformations (translations, rotations...) are also included in this definition.

2.2. Adversarial Examples

Adversarial perturbations are small artificial corruptions introduced into clean samples so as to fool deep neural networks. A lot of attacks can potentially harm most of the neural networks: [8, 2, 1]. In our study, we choose to consider four of the most broadly used attacks: FGSM, PGD, LL-FGSM and LL-PGD. They are efficient, and can be easily computed to test the neural network robustness.

FGSM (Fast Gradient Sign Method) is one of the simplest method used to build adversarial examples [8]:

$$x_{adv} = x + \epsilon * sign(\nabla_x L(x, l_{true})) \tag{1}$$

With x a sample to transform, l_{true} the corresponding label and L the cost function of the model. We note ϵ the amount of introduced adversarial perturbation.

PGD (Projected Gradient Descent) is an iterative version of FGSM [1]:

$$x^{k+1} = x^k + \frac{\epsilon}{n} * sign(\nabla_x L(x^k, l_{true})) \tag{2}$$

Starting with $x^0 = x$, the upper expression is computed n times to craft an adversarial example.

Instead of increasing the value of the loss function, it is possible to target a class in order to make a neural network associate the received sample with the targeted class. In particular, LL-FGSM (Least Likely FGSM) is a variation of FSGM that targets the class for which the targeted neural network has given the lowest score [1]. Considering l_{least}, the label corresponding to the lowest score given by a neural network, an adversarial example is crafted by computing the following formula:

$$x_{adv} = x - \epsilon * sign(\nabla_x L(x, l_{least})) \tag{3}$$

Similarily to LL-FGSM, LL-PGD is a variation of PGD that intends to make neural networks give a high score to the label l_{least} [1]. Starting with $x^0 = x$, the adversarial example is built by computing several times the following expression:

$$x^{k+1} = x^k - \frac{\epsilon}{n} * sign(\nabla_x L(x^k, l_{least})) \tag{4}$$

When the targeted model is known (we have access to its architecture and weights), we can directly use it to compute the gradient required for an adversarial attack. In this case, it is a white-box attack. These attacks are particularly harmful because they are built specifically to fool a precise model.

When the targeted model is unknown, the gradient used for the adversarial example crafting is computed with a different model. This is called a black-box attack. The model used for the gradient computing has a different architecture and different weights than the attacked model. Adversarial examples are transferable among neural networks [30]. It means that attacks built on a network generally fool other networks, even if they are very different. Then, black-box attacks remain harmful on most models. Making a neural network robust to any black-box attack is a challenging issue.

2.3. Data Augmentation

Data augmentation is a technique used to make a neural network more robust to a kind of perturbation. It consists in introducing corrupted samples into the training set. In other words, a neural network that is trained with data augmentation, learns on clean samples but also on samples modified with a perturbation. At the end of the training, the augmented neural network has been made robust to the perturbation used during the training [28]. When the perturbation used during the training is an adversarial attack, the data augmentation procedure is called adversarial training [8].

2.4. Robustness

The definition of robustness is not a consensus. It can refer to different concepts depending on the context. In this paper, the notion of robustness is considered in relation with some perturbations. We explicitly indicate the perturbations towards which the robustness is considered. For instance we study the robustness to translations or the robustness to adversarial examples etc.

We note A_{clean}, the accuracy of the neural network N on a test set. We consider some perturbations ϕ which are used to modify the samples of this test set. A_ϕ is the accuracy of the model on the test set modified with a ϕ perturbation. We measure the robustness of N to a ϕ perturbation with the expression:

$$R_N^\phi = \frac{A_\phi}{A_{clean}} \tag{5}$$

We call it a robustness score and it measures the accuracy loss due to the ϕ perturbation. The more the robustness score of a model is close to one, the more it is robust to the considered perturbation. To measure the robustness score of a neural network to a set of perturbations S, we use:

$$R_N^S = \sum_{\phi \in S} R_N^\phi \tag{6}$$

2.5. Related Works

Benchmark to estimate the robustness of neural networks.
In [11], the ImageNet-C benchmark is used to measure the robustness of neural networks to common perturbations. Their benchmark is built on a set of common perturbations on which neural networks should be tested. Unfortunately, some kinds of perturbations in the set are over-represented and some others are not taken into account. In particular, ImageNet-C contains three kinds of noises and four kinds of blurs, but occlusions, translations and rotations are not present in it. Then, we decided to build another benchmark, more representative of common perturbations encountered in real application cases. The method used to build it is given in Section 3.

The robustness to adversarial examples of a neural network, is usually measured by testing the performances of this network against several kinds of adversarial attacks [32, 16, 26].

Links between adversarial robustness and robustness to common perturbations.
A few works study the links between the adversarial robustness and some specific noise robustnesses. For instance, relations between adversarial perturbations and random noises are established in [7, 13]. These relations prove that neural networks can be robust to random noises and remain vulnerable to adversarial attacks.

Links between small geometric transformations and adversarial examples have been established in [34, 6]. It is argued that the robustness to additive adversarial perturbations and the robustness to rotations and translations are orthogonal concepts.

A few links between some common perturbation robustnesses and some adversarial attack robustnesses are established in these works. However, they compare adversarial robustness with the robustness to a few specific common perturbations. In this study, we consider robustness to common perturbations in a broad sense: most of the common perturbations are included. We want to know if the global adversarial robustness could help neural networks to be globally more robust to common perturbations and conversely. We intend to enlarge the scope of the previous works to know more about the way the neural network robustnesses are related to each other.

3. Construction of the Common Perturbation Benchmark

3.1. Experiment Set-up

We choose the ImageNet dataset to carry out our experiments [15]. ImageNet is widely used, challenging and big enough for achieving adversarial trainings [27]. Adversarial training and data augmentation are computationally expensive, they increase the training times of neural networks. To speed up the trainings, we decided to use a subset of ImageNet. It is composed of 5 super-classes, each regrouping several ImagetNet classes. The chosen classes are: *bird*, *dog*, *insect*, *primate* and *fish*. They correspond respectively to the ImageNet class ranges 80-100, 151-268, 300-319, 365-382 and 389-397. The choice of the classes was made by drawing inspiration from the experiments conducted in [33]. We insure the size equality of the classes by splitting the biggest classes to fit the smallest one. The resulting classes each contain ten thousand images. The obtained dataset is more suitable for achieving dozens of trainings in a reasonable amount of time, without loosing generality regarding the robustness study we want to conduct.

We use the ResNet-18 and ResNet-50 neural networks [10] for our studies. The results shown in the Tables of the paper have been obtained with ResNet-18. Yet, the same experiments conducted with ResNet-50 lead us to the same conclusions than the ones found with ResNet-18. Those models are trained to classify the images extracted from our ImageNet sub-dataset. We use a stochastic gradient descent with a learning rate of 0.01. The learning rate is divided by 10 when the training accuracy reaches a plateau. We use a weight decay of 0.0001 and a batch size of 128. The loss used is a cross-entropy function. We call the *stan-*

dard model, a ResNet-18 trained with these hyperparameters without using any data augmentation. It has an accuracy A_{clean} of 0.83 on the ImageNet sub-dataset. We also train a VGG network [29] and use it to get the gradient for the black-box adversarial attacks. For the trainings and the tests introducing adversarial examples, the amount of the corruption (ϵ) of each example, is randomly chosen. It varies from 0.01 to 0.1 for image pixel values that range from -1 to 1.

3.2. Perturbation Selection Criteria

To conduct the study, we need a method to estimate the robustness to common perturbations of neural networks. A natural way to do this is to estimate the network robustness to diverse kinds of common perturbations. The quality of the estimation greatly depends on the relevance of the chosen perturbations. We build a set of perturbations based on three selection criteria: completeness, virulence and non-overlapping.

Completeness. A complete set of common perturbations should cover most of the perturbations commonly encountered in real world applications. To be considered robust to common perturbations, a neural network should be robust to as many common perturbations as possible. To build the most exhaustive list of perturbations, we gather ideas from several sources. We choose perturbations encountered in various industrial applications: video surveillance, production line, autonomous driving, etc... We also get inspired by some other works [11, 14]. At this step, we obtain the following set of perturbations: Gaussian noise, salt-pepper noise, speckle noise, defocus blur, motion blur, zoom blur, glass blur, rotations, translations, vertical flips, obstructions, brightness variations, contrast loss, colorimetry variations, interference distortions, quantizations and jpeg compression.

The Gaussian noise may appear because of sensors high temperature or poor illumination during acquisition. Salt-pepper noise is generally due to errors caused by a conversion from an analog signal to a digital signal. Speckle noise often corrupts images captured by radars or medical imaging systems. Defocus blur appears because of bad camera focusing. Motion blur is caused by camera motions or displacement of observed objects. Zooms of cameras can introduce zoom blur. Glass blur is often observed because of translucent obstacles. The orientation and the position of observed objects can change depending on the context. For instance some pieces in a production line can be displaced or be inside out. We model this with translations, rotations and vertical flip transformations. Brightness, contrast and colorimetry vary with lightning conditions and sensors characteristics. Electrical interferences may appear during image capturing and perturb images. We model these interferences with small periodic artifacts. Quantization causes rounding errors that modify images. Some artifacts can appear because of jpeg compressions.

Virulence. Some perturbations are very virulent and disturb significantly neural networks. Some other corruptions are harmless: they do not cause a significative drop in model performances. Then, being robust to harmless perturbations is not an interesting attribute. We test the robustness of the *standard* model with all the perturbations selected in the previous paragraph. Most of these perturbations are virulent. For instance, the robustness score measured for Gaussian noise is 0.81. However, for quantizations or jpeg compression distortions, the measured robustness scores are above 0.97. We consider both corruptions not virulent. They are removed from the set of perturbations.

Non-overlapping. The robustnesses to two distinct perturbations can be correlated. Making a neural network more robust to a perturbation can also make it more robust to another perturbation and conversely. In this case, we consider that these robustnesses overlap. The presence of some overlapping perturbation robustnesses can unbalance the sum computed with the formula (5). If a kind of robustness is over-represented, it distorts the robustness measure.

We conduct experiments to show that the robustnesses to traditional noises overlap. We train three identical Resnet-18, respectively augmented with Gaussian noise, salt-pepper noise and speckle noise. For each model, its robustness score towards each noise is computed: results are presented in the left part of Table 1. Each line of the table except the standard one refers to an augmented model. Each column of the table refers to the noise used to compute the robustness scores of the line. For instance, the model augmented with a speckle noise, tested on samples corrupted with a salt-pepper noise, has a robustness score of 0.96. Every table of the paper is built this way: the lines refer to various models, the columns refer to the perturbations to which the robustnesses of the networks are evaluated.

In our noise study, we observe that if a model is robust to either the Gaussian noise, the salt-pepper noise or the speckle noise, it is also robust to the others. Then, these noise robustnesses overlap and so only the Gaussian noise is kept for the study.

Similarly, we train four identical Resnet-18, respectively augmented with defocus blur, zoom blur, motion blur and glass blur. Taking into account the robustness scores reported in the right part of Table 1, it appears that different kinds of blur robustnesses also overlap. Therefore, only the defocus blur is considered in further studies. We simply call it blur.

Blur and noise perturbations are the only perturbations of our list for which a robustness overlapping is observed.

Selected Perturbations. Relying on those three criteria, we finally gather the set of common perturbations shown in Figure 1. From the upper left image of this Figure to

Model \ Noise	gaus	salt	speck
standard	0.81	0.79	0.87
gaussian	0.98	0.98	0.98
salt_pepper	0.99	0.99	0.97
speckle	0.97	0.96	0.99

Model \ Noise	defo	zoom	glass	motion
standard	0.61	0.80	0.74	0.74
defocus_blur	0.98	0.94	0.98	0.94
zoom_blur	0.89	0.98	0.93	0.93
glass_blur	0.96	0.94	0.98	0.95
motion_blur	0.90	0.93	0.95	0.99

Table 1: Left. Robustness scores towards noise perturbations — Right. Robustness scores towards blur perturbations. Each line title of the table refers to the model used to compute the robustness score. The line "salt-pepper" for instance, corresponds to a model for which the training set has been augmented with salt-pepper corrupted images. Each column name refers to the perturbation used to compute the robustness scores of the column. We observe correlations between punctual noise robustnesses and correlations between blur robustnesses.

the lower right one, the corresponding perturbations may be abbreviated with: *gaus*, *art*, *obstr*, *blur*, *contr*, *bright*, *color*, *trans*, *rot* and *flip*.

3.3. Perturbation Intensities

Each of the selected perturbations except *flip* is associated with a range of intensity. For instance, square masks used for the occlusion perturbation can vary from 5 to 15 percent of the image size. We define a procedure to fix the upper and lower bounds of each perturbation intensity range. In order to fix the lower bound, the procedure starts with a very small perturbation. The intensity of the perturbation is progressively raised. During this increase, the behavior of the *standard* model on the corrupted images is periodically tested. The lower bound of the perturbation range is reached when the accuracy of the neural network starts to decrease. We keep increasing the severity until the perturbation becomes visually disturbing for humans. At this point, the upper bound is fixed.

To our knowledge, this is the first set of common perturbations built on a such thorough method of selection. It is sufficiently complete to cover most of the widely spread perturbations. It can be used either in a data augmentation procedure or to estimate the robustness to common perturbations of a neural network. It is small enough to be used without increasing the computational cost of trainings and tests too much. We believe that using this set of perturbations, could help to build neural networks more stable when deployed in real environments.

4. Common Perturbation Robustness and Adversarial Robustness

4.1. Robust Network Constructions

To study the links between the adversarial robustness and the robustness to common perturbations, it is necessary to build neural networks robust to common perturbations on the one side, and neural networks robust to adversarial examples on the other side.

A model robust to common perturbations. We first train a ResNet-18 augmented with all the common perturbations of our set. We call it the *fully augmented* model. We test its robustness to each perturbation of the set. The robustness scores obtained are compared with the ones obtained with the *standard* ResNet-18: results can be found in Table 2. As expected, the *fully augmented* model is much more robust to any tested perturbation than the *standard* one.

Data augmentation is supposed to make neural networks robust only to the distortions used in the augmentation process [5]. Then, there is no guarantee that the *fully augmented* model is robust to common perturbations on which it has not been trained. We build a second experiment to guarantee that the *fully augmented* network is more robust to any common perturbations than the *standard* model. To achieve this, we train several ResNet-18, each augmented with all perturbations of our set but one. For instance, the *no-gaussian* model is the model augmented with all perturbations but the Gaussian noise. In a general way, the *no-*ϕ model has been submitted to all perturbations of the set but ϕ. We compute the robustness of each *no-*ϕ model on samples corrupted with the ϕ perturbation. We compare every robustness score found this way with the ones of the *standard* model: results are presented in Table 3.

It appears that even if each *no-*ϕ model has never seen the ϕ perturbation, it is slightly more robust to it than the *standard* model. It is true that the robustnesses to two very different distortions are usually not correlated: robustness to blur does not help with Gaussian noise. Yet, we observe that an increase in robustness to a group of common perturbations, can imply a better robustness to a very different perturbation. The *no-gaussian* model is more robust to Gaussian noise than the *standard* model.

Therefore, with a sufficiently large and diverse set of

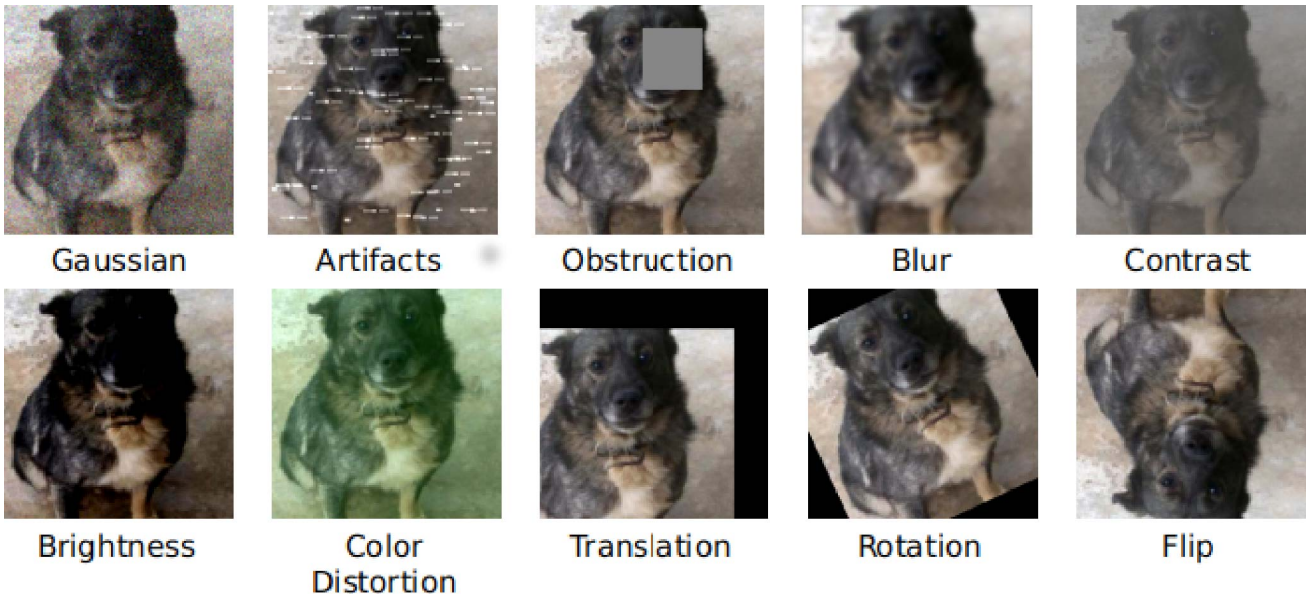

Figure 1: Visualization of the selected common perturbations for the benchmark. These perturbations are fundamentally different by nature and affect distinct characteristics of images. They cannot be reduced to a smaller set without a significant loss of diversity. Each perturbation except *flip* is provided with a continuous range of severity.

Noise / Model	gaus	art	obstr	blur	contr	bright	color	trans	rot	flip	mean
standard	0.81	0.81	0.95	0.78	0.87	0.94	0.72	0.89	0.94	0.78	0.85
fully augmented	0.96	0.98	0.97	0.95	0.98	0.97	0.96	0.98	0.98	0.92	0.97

Table 2: Efficiency of data augmentation on robustness. The robustness scores of the *standard* and *fully augmented* models are computed on the perturbations of the benchmark. Data augmentation makes the *fully augmented* model much more robust to common perturbations than the *standard* model.

common perturbations, it is possible to make a neural network more robust to an unseen common perturbation. As the *fully augmented* model is trained on more perturbations than *no-*ϕ models, it has even more chances to be robust to any common perturbations. Consequently, in further experiments, the *fully augmented* model is considered globally more robust to common perturbations than the *standard* model.

A model robust to adversarial examples. To build a ResNet-18 robust to adversarial attacks, we use adversarial training. We call *fgsm*, *pgd*, *ll_fgsm* and *ll_pgd*, the models respectively augmented with FGSM, PGD, LL-FGSM and LL-PGD adversarial examples. To estimate their adversarial robustness, we compute their robustness scores on every attack introduced in section 2.2. These scores are computed in black-box and white-box settings. Results are provided in Table 4. Compared to the *standard* model, the adversarially trained models are more robust to every adversarial attack we test, in black-box and white-box settings.

Besides, even if each of these models has been trained only with one kind of adversarial example, all of them are relatively robust to other kinds of adversarial examples. For instance the *fgsm* model is more robust than the *standard* model to LL-FGSM, PGD and LL-PGD adversarial examples. It means that adversarial example robustnesses are correlated: making a network robust to a specific adversarial attack, helps it to deal with other kinds of adversarial perturbations.

4.2. Links between Adversarial Robustness and Common Perturbation Robustness

We can observe correlations between some common perturbation robustnesses. Likewise, increasing the robustness to an adversarial attack makes neural networks less sensitive to other adversarial perturbations. But are there correlations between adversarial robustnesses and robustness to

Model \ Noise	gaus	art	obstr	blur	contr	bright	color	trans	rot	flip
standard	0.81	0.81	0.95	0.78	0.87	0.94	0.72	0.89	0.94	0.78
no-ϕ	0.83	0.84	0.96	0.82	0.94	0.97	0.76	0.92	0.95	0.78

Table 3: Robustness to unseen perturbations. Each score of the second line refers to the robustness of a *no-ϕ* model against the ϕ perturbation. Various augmentations help the *no-ϕ* models to deal with the unseen ϕ perturbations.

Model \ Attack	fgsm	fgsm_ll	pgd	pgd_ll
standard	0.68	0.70	0.73	0.95
fgsm	0.96	0.97	0.97	99
fgsm_ll	0.96	0.97	0.98	0.99
pgd	0.98	0.98	0.98	1.00
pgd_ll	0.93	0.94	0.98	0.99
fully augmented	0.67	0.68	0.73	0.95

Model \ Attack	fgsm	fgsm_ll	pgd	pgd_ll
standard	0.02	0.07	0.00	0.04
fgsm	0.62	0.89	0.27	0.67
fgsm_ll	0.42	0.75	0.36	0.79
pgd	0.47	0.86	0.42	0.85
pgd_ll	0.41	0.79	0.38	0.82
fully augmented	0.02	0.08	0.01	0.05

Table 4: Left. Robustness to black-box attacks — Right. Robustness to white-box attacks
The columns of the tables refer to the adversarial attacks used to perturb a model, either in a black-box (left) configuration or in a white-box configuration (right). The adversarially trained models are much more robust to adversarial examples than the *standard* and *fully augmented* models.

common perturbations ?

We measure the robustness of the network augmented with common perturbations to adversarial examples and vice-versa. In Table 4, it appears that the *fully augmented* model is not more robust than the *standard* model to adversarial attacks: their robustness scores are almost equal. So, robustness to common perturbations does not protect from adversarial attacks. Similarly, as showed in Table 5, the adversarially trained models are not more robust to the common perturbations than the *standard* model. Increasing the robustness to adversarial examples does not increase the robustness to common perturbations. Therefore, adversarial robustness and robustness to common perturbations are independent attributes.

5. Discussions

We showed that relations exist between robustnesses to common perturbations. It also exists correlations between adversarial examples robustnesses. However adversarial robustness and common perturbation robustness are not correlated.

This discrepancy could be explained by the significant difference of nature between adversarial perturbations and common perturbations. An adversarial perturbation is an attack. It is made to disturb neural networks and it depends on the sample it corrupts: see formula (1). Adversarial examples are based on an addition procedure and they are made with very small perturbations. On the other hand, common perturbations do not adapt to the neural networks they affect. They are not necessarily additive and can be very severe. Therefore, the way common perturbations and adversarial perturbations affect neural networks might be drastically different.

Recent works show the difference of nature between two kinds of features of images called *robust* and *non-robust features* [12]. The *none-robust features* are the features exploited by adversarial examples to disturb the neural networks they attack. The *robust features* are the features that are not modified by adversarial perturbations. In [12], it is shown that adversarially trained models rely on *robust features* while the models not augmented with adversarial perturbations rely on *non-robust features* in addition to the *robust features*. We think the differences of features used by these networks could explain the independence between adversarial robustness and robustness to common perturbations.

Be that as it may, robustness cannot be reduced to the sole adversarial robustness or the sole robustness to common perturbations. Those robustnesses are too poorly related. This independence is restrictive for industrial applications. Both robustnesses have to be addressed and addressed independently. It means that it would be necessary to introduce more regularization techniques or diversify the samples used in the augmentation procedures. This solution

Model \ Noise	gaus	art	obstr	blur	contr	bright	color	transl	rot	flip	mean
standard	0.81	0.81	0.95	0.78	0.87	0.94	0.72	0.89	0.94	0.78	0.85
fgsm	0.91	0.89	0.93	0.91	0.78	0.87	0.69	0.82	0.91	0.78	0.85
fgsm_ll	0.96	0.87	0.95	0.91	0.73	0.88	0.67	0.83	0.92	0.78	0.85
pgd	0.95	0.88	0.94	0.94	0.75	0.87	0.69	0.81	0.91	0.79	0.85
pgd_ll	0.94	0.85	0.95	0.93	0.75	0.86	0.61	0.83	0.92	0.79	0.84

Table 5: Robustnesses of adversarially trained models to common perturbations. The adversarially trained models are not more robust to common perturbations than the *standard model*.

is time-consuming and computationally expensive.

From a theoretical point of view, we expect robust neural networks to extract relevant features. These features should be stable and should not change with common perturbations or adversarial attacks. Then, methods to make neural networks more robust should not depend on the nature of the corruption. They should naturally cover most common perturbations and adversarial attacks at the same time. Then, we believe that neural network robustness should be addressed in a broader sense: covering adversarial examples, common perturbations or even unexpected kinds of distortions simultaneously.

A possible approach to address robustness in a broad sense is to use alternative adversarial example definitions. Some new formulations have been proposed to enlarge the scope of adversarial perturbations [6], [19]. These definitions include small rotations, translations or lightning variations into the adversarial attack scope. Using more general adversarial examples in trainings could be a way to erase the discrepancy between the adversarial robustness and the robustness to common perturbations. If we can find a wide enough definition of adversarial attacks, both robustnesses could be addressed simultaneously.

Generative Adversarial Networks have been used to increase the adversarial robustness of some neural networks [17]. A generator is used to submit disturbing samples to another trained neural network. The generator is supposed to target the weaknesses of the other model. It automatically finds relevant attacks that help the other neural network to increase its robustness. The advantage of GAN is that the allowed range of perturbations is almost unlimited. Perturbations introduced by the generator are not made by hand: they are not restricted to a few common perturbations or to some adversarial attacks. Generated attacks can contain a complex mix of common perturbations and adversarial patterns. GAN should be a good solution to introduce automatically a lot of kinds of perturbations in order to address robustness in a broader sense.

6. Conclusion

We carried out original experiments to better understand the links between the neural network robustnesses to different kinds of perturbations. We propose a new benchmark to estimate robustness to common perturbations. We showed that using data augmentation with a carefully chosen set of common perturbations, can increase the robustness of a model to an unknown common perturbation. We also demonstrated that adversarial robustness and robustness to common perturbations are independent attributes. We believe that the key to address neural network robustness in a broad sense, is to enlarge the scope of the perturbations used in trainings and tests, by considering corruptions that are in between common perturbations and adversarial attacks.

References

[1] Samy Bengio Alexey Kurakin, Ian J. Goodfellow. Adversarial examples in the physical world. In *International Conference on Learning Representations*, 2017.

[2] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, May 2017.

[3] S. Dodge and L. Karam. Understanding how image quality affects deep neural networks. *Eighth International Conference on Quality of Multimedia Experience (QoMEX)*, 2016.

[4] S. Dodge and L. Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7, July 2017.

[5] Samuel F. Dodge and Lina J. Karam. Quality resilient deep neural networks. *CoRR*, abs/1703.08119, 2017.

[6] Logan Engstrom, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. A rotation and a translation suffice: Fooling cnns with simple transformations. *CoRR*, abs/1712.02779, 2017.

[7] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural In-*

formation Processing Systems 29, pages 1632–1640. Curran Associates, Inc., 2016.

[8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2015.

[9] Shixiang Gu and Luca Rigazio. Towards deep neural network architectures robust to adversarial examples. *International Conference on Learning Representations*, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, June 2016.

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

[12] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *ArXiv*, abs/1905.02175, 2019.

[13] Omar Fawzi Jean-Yves Franceschi, Alhussein Fawzi. Robustness of classifiers to uniform l_p and gaussian noise. *International Conference on Artificial Intelligence and Statistic*, 2018.

[14] S. Karahan, M. Kilinc Yildirum, K. Kirtac, F. S. Rende, G. Butun, and H. K. Ekenel. How image degradations affect deep cnn-based face recognition? In *2016 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5, Sep. 2016.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.

[16] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017.

[17] Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. Generative adversarial trainer: Defense to adversarial perturbations with GAN. *CoRR*, abs/1705.03387, 2017.

[18] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014.

[19] Hsueh-Ti Derek Liu, Michael Tao, Chun-Liang Li, Derek Nowrouzezahrai, and Alec Jacobson. Beyond pixel norm-balls: Parametric adversaries using an analytically differentiable renderer. In *International Conference on Learning Representations*, 2019.

[20] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *ECCV*, 2018.

[21] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3730–3738, Dec 2015.

[22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

[23] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 582–597, May 2016.

[24] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39, 06 2015.

[25] Andrew Slavin Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *AAAI*, 2018.

[26] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *International Conference on Learning Representations*, 2018.

[27] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5014–5026. Curran Associates, Inc., 2018.

[28] P. Y. Simard, D. Steinkraus, and J. C. Platt. Best practices for convolutional neural networks applied to visual document analysis. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, pages 958–963, Aug 2003.

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

[30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[31] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, June 2014.

[32] Florian Tramr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018.

[33] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

[34] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *International Conference on Learning Representations*, 2018.

[35] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[36] Y. Zhou, S. Song, and N. Cheung. On classification of distorted images with deep convolutional neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1213–1217, March 2017.