Benchmarking the Corruption and Structural Robustness of Neural Networks

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

In this paper we establish rigorous benchmarks for image classifier robustness. Our first benchmark IMAGENET-C standardizes and expands the corruption robustness topic, while showing which classifiers are preferable in safety-critical applications. Unlike recent robustness research, this benchmark evaluates performance on commonplace not worst-case adversarial corruptions. We find that there are negligible changes in relative corruption robustness from AlexNet to ResNet classifiers, and we discover ways to enhance corruption robustness. Then we propose a new dataset called ICONS-50 which opens research on a new kind of robustness, structural robustness. With this dataset we evaluate the frailty of classifiers on new styles of known objects and unexpected instances of known classes.

1 Introduction

The use of deep neural networks [Krizhevsky et al., 2012] in pattern recognition systems has led to large advances on numerous benchmark tasks. However, vast gains on the sanitized test set belie an underlying fragility of these models: they are not robust to common input corruptions nor unexpected situations that we anticipate them to face in real-world. When tasked with a visual classification problem, humans tend to be robust to numerous natural and unnatural corruptions, such as snow, blur, and pixelation. Moreover, the introduction of novel corruptions and variations on existing corruptions tends not to change this fact. Humans also can observe new species of birds without utter confusion, indicating a robustness not just to image corruptions but also to abstract structural changes. It is also true that humans can withstand minute adversarial input perturbations [Szegedy et al., 2013, Carlini and Wagner, 2017, 2016a], but in this work we focus structural robustness and robustness to common corruptions, not robustness to pathological and worst-case adversarial corruptions. Endowing neural network-based systems with greater robustness is hence an important direction for bringing the performance of deep learning systems closer to that of humans, as well as for improving the safety and security of these systems for deployment in real-world environments like foggy roads.

To this end, first we propose a set of 75 common visual corruptions as a means of benchmarking the general corruption robustness of deep neural network image classifiers. With this benchmark called IMAGENET-C, we attempt to standardize corruption research to prevent incomparable evaluations, moving goalposts, and experiment cherry-picking. At the same time, IMAGENET-C enables further classification research since there is wide room for improvement on this challenging benchmark. In extensive experiments, we demonstrate the shortcomings of current systems on this benchmark, and find methods and architectures which make tangible progress on differentially improving robustness while retaining accuracy. We then turn to structural robustness, a robustness goal parallel to corruption robustness which has not yet received proper treatment. We lay groundwork in this domain by introducing a novel dataset called ICONS-50, enabling a series experiments to benchmark robustness to structural changes like new styles and novel animal species. These benchmarks make comprehensive robustness comparisons possible and point the way to more robust classifiers.

Submitted to 32nd Conference on Neural Information Processing Systems (NIPS 2018). Do not distribute.

2 Related Work

Adversarial Examples. An adversarial image is a clean image perturbed by a small, carefully crafted corruption so as to confuse a classifier [Szegedy et al., 2013]. These deceptive corruptions can occasionally fool black-box classifiers [Kurakin et al., 2017], and they approximate the smallest image modification in RGB space necessary for classifier confusion [Carlini et al., 2017]. Thus adversarial corruptions serve as type of worst-case analysis for network robustness. Its eminence has often led "adversarial robustness" to become interchangeable with "robustness" [Bastani et al., 2016, Rauber et al., 2017]. Several competing robustness measures are scattered throughout the literature [Bastani et al., 2016, Carlini and Wagner, 2016b, Carlini et al., 2017, Madry et al., 2017, Katz et al., 2017], but we find no indication of widespread adoption of one measure. New defenses [Lu et al., 2017, Papernot et al., 2017, Metzen et al., 2017, Hendrycks and Gimpel, 2016] quickly succumb to new attacks [Evtimov et al., 2017, Carlini and Wagner, 2017, 2016a]. Defending against adversaries and the worst-case can be useful, but generalizing to plain corruptions is already challenging.

Robustness in Speech. Speech recognition research approaches robustness differently [Li et al., 2014, Mitra et al., 2017]. Common audio corruptions (e.g., street noise, background chatter, wind) receive greater focus than adversarial audio, because common corruptions are ever-present and unsolved. There are several popular datasets containing noisy test audio [Hirsch and Pearce, 2000, Hirsch, 2007]. Robustness in noisy environments requires robust architectures, and some research finds convolutional networks more robust than fully connected networks [Abdel-Hamid et al.]. Additional robustness stems from pre-processing techniques like standardizing the input's statistics [Liu et al., 1993, Ángel de la Torre et al., 2005, Harvilla and Stern, 2012, Kim and Stern, 2016].

ConvNet Fragility Studies. Several studies demonstrate the fragility of convolutional networks on simple corruptions. For example, Hosseini et al. [2017] use impulse noise to break Google's Cloud Vision API. Using Gaussian noise and blur, Dodge and Karam [2017b] demonstrate the superior robustness of human vision to convolutional networks, *even after networks are fine-tuned* on Gaussian noise or blur. Geirhos et al. [2017] also compare networks to humans with noisy and elastically deformed images. They find that fine-tuning on specific corruptions does not generalize, and classification error patterns underlying network and human predictions are not similar.

Robustness Enhancements. In an effort to dispel classifier fragility, Vasiljevic et al. [2016] fine-tune on blurred images. They find it is not enough to fine-tune on one type of blur to generalize to other blurs, and fine-tuning on several blurs can marginally decrease performance. Zheng et al. [2016] also find that fine-tuning on noisy images can cause underfitting, so they encourage the noisy image softmax distribution to match the clean image softmax. Dodge and Karam [2017a] address underfitting differently. They fine-tune each network on one corruption and classify with an mixture of these corruption-specific experts, though they do not assess combinations of known corruptions.

3 The ImageNet-C Corruption Robustness Benchmark

3.1 The ImageNet-C Dataset

ImageNet-C Design. Our corruption robustness benchmark consists of 15 diverse corruption types, exemplified in Figure 1. The benchmark covers noise, blur, weather, and digital categories. Research that improves performance on this benchmark strongly indicates general robustness gains, as the corruptions are varied and great in number. These 15 corruption types each have five different levels of severity since corruptions can manifest themselves at varying intensities. Appendix A gives an example of a corruption type's five different severities. Real-world corruptions also have variation even at a fixed intensity. To simulate these, we introduce variation for each corruption when possible. For example, each fog cloud is unique to each image. These algorithmically generated corruptions are applied to ImageNet [Deng et al., 2009] validation images, giving rise to our corruption robustness dataset ImageNet-C. The dataset can be downloaded or re-created by visiting [anonymized]. Our benchmark tests networks with ImageNet-C images, *but networks do not train on these images*. Networks are trained with datasets such as ImageNet but not ImageNet-C. To enable further experimentation, we also designed an extra corruption for each noise, blur, weather, or digital category. Extra corruptions are depicted and explicated in Appendix B and also available at the aforementioned URL. Overall, the dataset ImageNet-C consists of 15 corruption types, each with five different severities, all applied to ImageNet validation images for testing a pre-existing network.

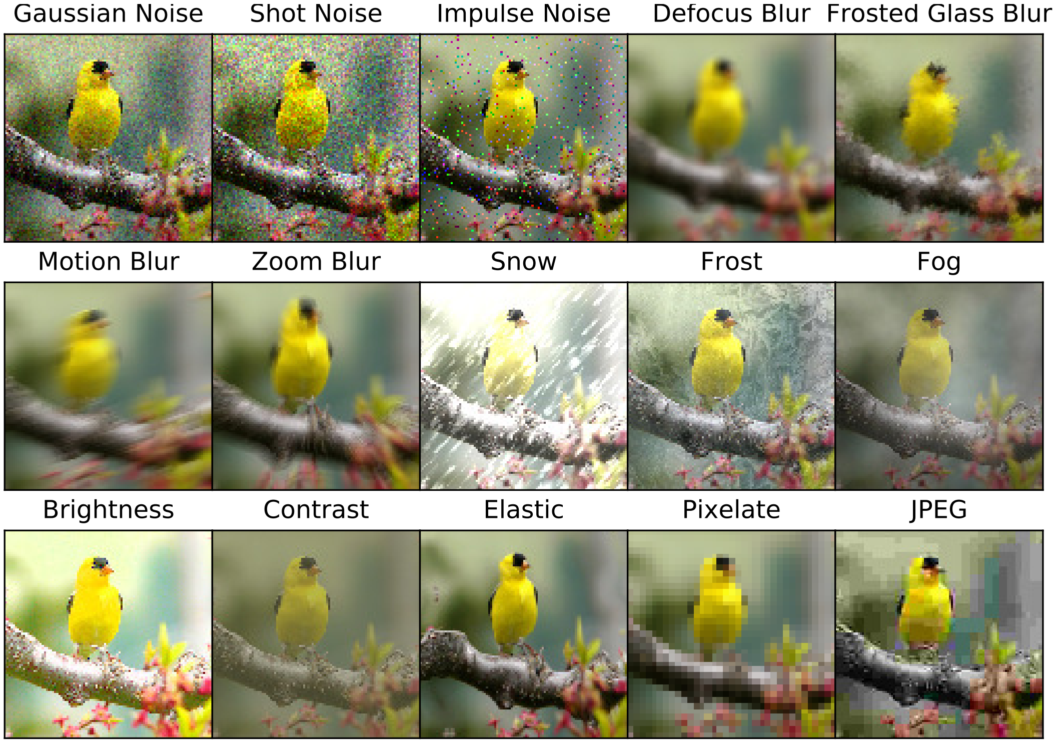

Figure 1: Our IMAGENET-C dataset consists of 15 types of algorithmically generated corruptions from noise, blur, weather, and digital categories. Each type of corruption has five levels of severity, resulting in 75 distinct corruptions. See different severity levels in Appendix A.

Common Corruptions. The first IMAGENET-C corruption is *Gaussian noise*. This corruption can appear in low-lighting conditions. *Shot noise*, also called Poisson noise, is electronic noise caused by the discrete nature of light itself. *Impulse noise* is a color analogue of salt-and-pepper noise and can be caused by bit errors. *Defocus blur* occurs when an image is out of focus. *Frosted Glass Blur* appears with "frosted glass" windows or panels. *Motion blur* appears when a camera is moving quickly. *Zoom blur* occurs when a camera moves toward an object rapidly. *Snow* is a visually obstructive form of precipitation. *Frost* forms when lenses or windows are coated with ice crystals. *Fog* shrouds objects and is rendered with the diamond-square algorithm. *Brightness* varies with daylight intensity. *Contrast* can be high or low depending on lighting conditions and the photographed object's color. *Elastic* transformations stretch or contract small image regions. *Pixelation* occurs when upsampling a low-resolution image. *JPEG* is a lossy image compression format that increases image pixelation and introduces artifacts. Each corruption type is tested with depth due to its five severity levels, and this broad range of corruptions allow us to test model corruption robustness with breadth.

3.2 Metric and Setup

Mean and Relative Corruption Error. Common corruptions such as Gaussian noise can be benign or destructive depending on their severity. In order to *comprehensively* evaluate a classifier's robustness to a given type of corruption, we score the classifier's performance across five corruption severity levels and aggregate its scores. The first evaluation step is to take a pre-existing classifier here notated "Network," which has not and will not train on IMAGENET-C, and then compute the clean dataset top-1 error rate. Denote this error rate $E^{\text{Network}}_{\text{Clean}}$. This same classifier will then test on an IMAGENET-C corruption type notated "Corruption." Let top-1 error rate for the Network classifier on Corruption with severity level s $(1 \leq s \leq 5)$ be written $E^{\text{Network}}_{s,\text{Corruption}}$. The classifier's aggregate performance across the five severities of a corruption type is the Corruption Error, computed

$$\text{CE}^{\text{Network}}_{\text{Corruption}} = \sum_{s=1}^{5} E^{\text{Network}}_{s,\text{Corruption}} \bigg/ \sum_{s=1}^{5} E^{\text{AlexNet}}_{s,\text{Corruption}}.$$

Observe that we normalize by AlexNet's errors in order to normalize by the corruption's difficulty. Fog corruptions often obscure an object's class more than Brightness corruptions. Thus to make Corruption Errors comparable across corruption types, we adjust for the difficulty by dividing by AlexNet's errors. Now with commensurate Corruption Errors, we can summarize model corruption robustness by averaging the 15 Corruption Error values $\left(\mathrm{CE}^{\text{Network}}_{\text{Gaussian Noise}}, \mathrm{CE}^{\text{Network}}_{\text{Shot Noise}}, \dots, \mathrm{CE}^{\text{Network}}_{\text{JPEG}}\right)$. This results in the *mean CE* or *mCE* for short.

We now introduce a more nuanced corruption robustness measure. Consider a classifier that withstands most corruptions, so that the gap between the mCE and the clean data error is minuscule. Contrast this with a classifier with a low clean error rate but nonetheless does not cope well with corruptions, corresponding to a large gap between the mCE and clean data error. It is possible the former classifier has a larger mCE than the latter, despite the former degrading more gracefully in the presence of corruptions. The amount that the classifier declines on corrupted inputs is given by the formula Relative $\mathrm{CE}^{\text{Network}}_{\text{Corruption}} = \left(\sum_{s=1}^{5} E^{\text{Network}}_{s,\text{Corruption}} - E^{\text{Network}}_{\text{Clean}}\right) / \left(\sum_{s=1}^{5} E^{\text{AlexNet}}_{s,\text{Corruption}} - E^{\text{AlexNet}}_{\text{Clean}}\right)$. Averaging these 15 Relative Corruption Errors results in the *Relative mCE*. In short, the Relative mCE measures the relative robustness or the performance degradation when encountering corruptions.

Preserving Metric Validity. Keeping both metrics valid requires that researchers not directly train on any of these 75 corruptions. The ImageNet-C dataset is designed for testing not training networks. To reduce implicitly fitting these corruptions, we provide extra corruptions with which to validate models, discussed in Appendix B. Fine-tuning a model on each corruption is not in the spirit of generalization to new and unexpected settings. Note that demanding generalization to a novel corruption is reasonable since, for example, humans can generalize to new Instagram filters with ease. Additionally, the realm of corruptions for fine-tuning at least includes uniform noise, median blur, fisheye lens distortion, and exponentially many combinations of other corruptions. What is more is that fine-tuning on specific corruptions tends not to provide generalization to new corruptions, and models tuned and tested on a specific corruption still remain below humans (see Related Work). For these reasons, ImageNet-C corruptions should remain unseen until test time.

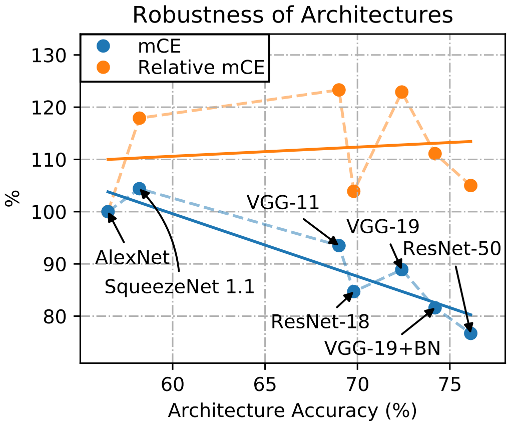

Figure 2: Robustness (mCE) and Relative mCE ImageNet-C values. Relative mCE values suggest robustness in itself declined from AlexNet to ResNet. "BN" abbreviates Batch Normalization.

3.3 Architecture Robustness

Have architectures become more robust since AlexNet [Krizhevsky et al., 2012]? As seen in Figure 2, as architectures improve, so too does the mean Corruption Error (mCE). By this measure, architectures have become progressively more successful at generalizing to corrupted distributions. All Corruption Error values are in Appendix C. Note that models with similar clean error rates have fairly similar CEs, and there are no large shifts in a corruption type's CE. Consequently, it would seem that architectures have slowly and consistently improved their representations over time. However, it appears that robustness improvements are mostly explained by accuracy improvements. Recall that the Relative mCE tracks a classifier's accuracy decline in the presence of corruptions. Figure 2 shows that the Relative mCE is worse than that of AlexNet. In consequence, from AlexNet to ResNet [He et al., 2015], robustness in itself has barely changed. Relative robustness remains near AlexNet-levels and therefore beneath humans, revealing our "superhuman" classifiers to be decidedly subhuman.

3.4 Informative Robustness Enhancement Attempts

Stability Training. Stability training is a technique to improve the robustness of deep networks [Zheng et al., 2016]. The method's creators found that training on images corrupted with noise can lead to underfitting, so they instead propose minimizing the cross-entropy from the noisy image's softmax distribution to the softmax of the clean image. The authors evaluated its performance on images with subtle differences and suggested that the method provides additional robustness to JPEG corruptions. We fine-tune a ResNet-50 with stability training for five epochs. For train time corruptions, we corrupt images with uniform noise, where the maximum and minimum of the uniform noise is tuned over $\{0.01, 0.05, 0.1\}$, and the stability weight is tuned over $\{0.01, 0.05, 0.1\}$. Across all noise strengths and stability weight combinations, the model with stability training tested on IMAGENET-C had a larger mCE than the baseline ResNet-50's mCE. Even on unseen noise corruptions, stability training did not increase robustness. An upshot of this failure is that benchmarking robustness-enhancing techniques requires a diverse test set.

Image Denoising. An approach orthogonal to modifying model representations is to improve the inputs using image restoration techniques. Although *general* image restoration techniques are not yet mature, denoising restoration techniques are not. We thus attempt restore an image with the denoising technique called non-local means [Buades and Coll, 2005]. The amount of denoising applied is determined by the noise estimation technique of Donoho and Johnstone [1993]. Therefore clean images receive nearly no modifications from the restoration method, while noisy images should undergo considerable restoration. But for all that effort, denoising slightly increased the mCE. A plausible account is that the non-local means algorithm slightly smoothed images even when images lacked noise, despite having the non-local means algorithm governed by the noise estimate. Therefore, the gains in noise robustness were wiped away by subtle blurs to images with other types of corruptions, showing that targeted image restoration may prove harmful for robustness.

Smaller Models. All else equal, "simpler" models often generalize better, and "simplicity" frequently translates to model size. Accordingly, smaller models may be more robust. We test this hypothesis with CondenseNets [Huang et al., 2017b]. A CondenseNet affords its small size by virtue of sparse convolutions and pruned filter weights. An off-the-shelf CondenseNet ($C = G = 4$) obtains a 26.3% error rate and a 80.8% mCE. On the whole, this CondenseNet is slightly less robust than larger models of similar accuracy. Even more pruning and sparsification yields a CondenseNet ($C = G = 8$) with both deteriorated performance (28.9% error rate) and robustness (84.6% mCE). Here again robustness is worse than larger model robustness. Though models fashioned for mobile devices are smaller and in some sense simpler, this is not enough for robustness gains let alone its preservation.

3.5 Successful Robustness Enhancements

Histogram Equalization. Histogram equalization successfully standardizes speech data for robust speech recognition [Ángel de la Torre et al., 2005, Harvilla and Stern, 2012]. For images, we find that preprocessing with Contrast Limited Adaptive Histogram Equalization [Pizer et al., 1987] is quite effective. Unlike our previous image denoising attempt, CLAHE reduces the effect of some corruptions while not worsening performance on most others, thereby improving the mCE. We demonstrate CLAHE's net improvement by taking a pretrained ResNet-50 and fine-tune the whole model for five epochs on images processed with CLAHE. The ResNet-50 has a 23.87% error rate, but ResNet-50 with CLAHE has an error rate of 23.55%. On nearly all corruptions, CLAHE slightly decreases the Corruption Error. The ResNet-50 without CLAHE preprocessing has an mCE of 76.7%, while with CLAHE the ResNet-50's mCE decreases to 74.5%.

Multiscale, Feature Aggregating, and Larger Networks. Multiscale architectures achieve greater robustness by propagating features across scale at each layer rather than slowly gaining a global representation of the input as in typical convolutional neural networks. Some multiscale architectures are called Multigrid Networks [Ke et al., 2016]. Multigrid networks each have a pyramid of grids in each layer which enables the subsequent layer to operate across scales. Along similar lines, Multi-Scale Dense Networks (MSDNets) [Huang et al., 2017a] use information across scales. Distinctly, MSDNets bind network layers with DenseNet-like [Huang et al., 2017c] skip connections. These two different multiscale networks both enhance robustness. Before comparing mCE values, we first note the Multigrid network has 24.6% top-1 error, as does the MSDNet, while the ResNet-50 has 23.9% top-1 error. On noisy inputs, Multigrid networks noticably surpass ResNets and MSDNets, as shown in Figure 3. Since these multiscale architectures have high-level representations processed in tandem with fine details, the architectures appear better equipped to suppress otherwise distracting pixel noise. When all corruptions are evaluated, ResNet-50 has an mCE of 76.7%, the MSDNet has an mCE of 73.6%, and the Multigrid network has an mCE of 73.3%.

Some recent models enhance the ResNet architecture by increasing what is called feature aggregation. Of these, DenseNets [Huang et al., 2017c] and ResNeXts [Xie et al., 2016] are most prominent.

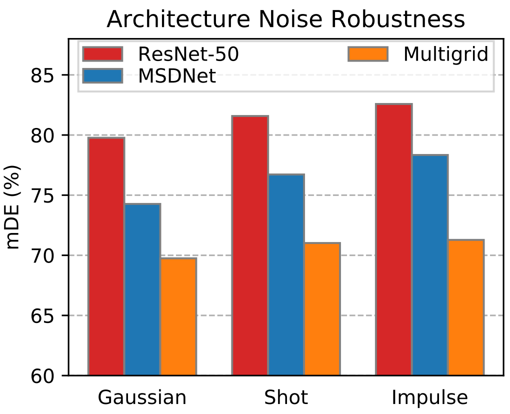

Figure 3: Architectures like Multigrid networks operate on representations across different scales and can more effectively resist noise corruptions.

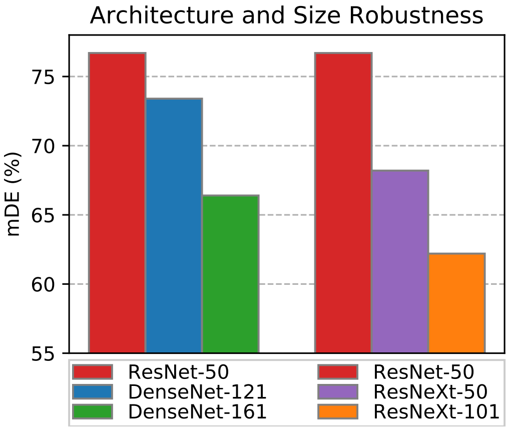

Figure 4: Moving to larger networks can generate robustness gains that substantially outpace accuracy gains.

Each purports to have stronger representations than ResNets, and the evidence is largely a hard-won ImageNet error-rate downtick, but our robustness measure is less saturated. So, the ImageNet-D mCE clearly indicates that DenseNets and ResNeXts have superior representations. Accordingly, a switch from a ResNet-50 (23.9% top-1 error) to a DenseNet-121 (25.6% error) decreases the mCE from 76.7% to 73.4%. More starkly, switching from a ResNet-50 to a ResNeXt-50 (22.9% top-1) drops the mCE from *76.7% to 68.2%*. Results are summarized in Figure 4. We discovered that the top-1 ImageNet error reductions underrated but ImageNet-C appropriately corroborated the higher caliber of DenseNet and ResNeXt representations.

Some of the greatest and simplest robustness gains sometimes emerge from making recent models more monolithic. Apparently more layers, more connections, and more capacity allow these massive models to operate more stably on corrupted inputs. We saw earlier that seen making models smaller does the opposite. Swapping a DenseNet-121 (25.6% top-1) with the larger DenseNet-161 (22.9% top-1) decreases the mCE from 73.4% to 66.4%. In a similar fashion, a ResNeXt-50 (22.9% top-1) is less robust than the a giant ResNeXt-101 (21.0% top-1) since the mCEs are 68.2% and 62.2% respectively. Both model size and feature aggregation results are summarized in Figure 4. Consequently, future models with more depth, width, and feature aggregation may give way to further robustness.

4 Structural Robustness

In addition to testing for robustness to image-level corruptions, we want to test robustness to high-level image structure changes. To accomplish this, we created a second new dataset called Icons-50. It enables testing two types of high-level structural changes. The first is *style robustness*, or robustness to how an object is represented. The second is *subclass robustness*, or robustness to unexpected instances of a class. We present Icont-50 then test structural robustness in many contexts.

4.1 The Icons-50 Dataset

The new Icons-50 dataset features icons of animals, people, food, activity, places, objects, and symbols, for a total of 50 classes and 10,000 images. A full list of Icons-50 classes is in Appendix E, and the dataset can be downloaded at [anonymized]. A subset of Icons-50 classes is shown in Figure 5. Each class has icons with different styles. For instance, icons with the thick, black outlines (like the bottom right "Drink" icon) are stylized by Microsoft for their operating systems. Other styles in the Icons-50 dataset are from Apple, Samsung, Google, Facebook, and other platforms.

These various styles supply great dataset density, or numerous examples per class. Icons-50 density exceeds that of logo datasets, the most comparable datasets. The dataset of Romberg et al. [2011] has approximately 70 logos per class and 32 classes. Likewise, the dataset of Joly and Buisson [2009] has approximately 53 logos per class and 37 classes. Contrariwise, Icons-50 has 200 icons per class on average and 50 classes. Relative to logo datasets, Icons-50 has up to five times the density.

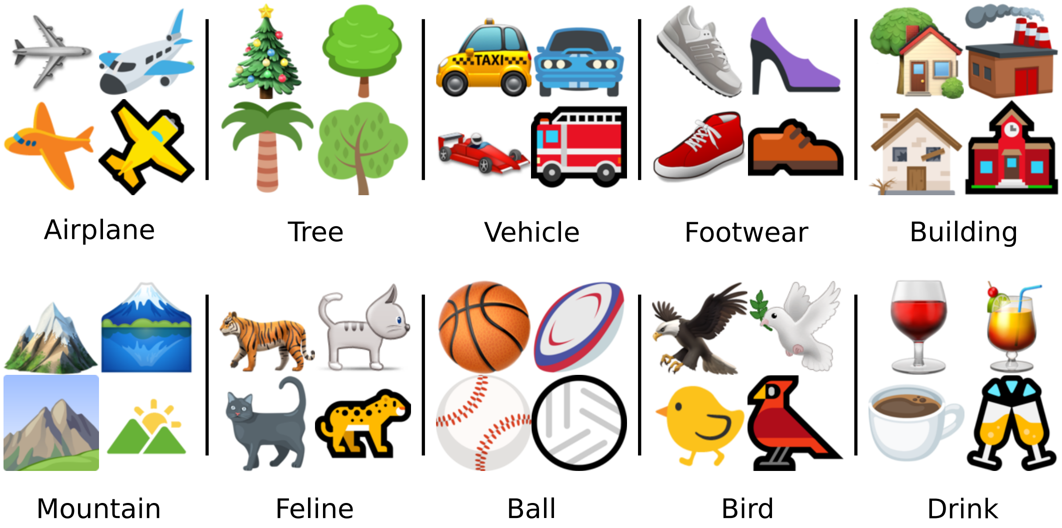

Figure 5: Image samples are from 12 of the 50 ICONS-50 classes. These images showcase the dataset's high image quality, interclass and intraclass diversity, and its many styles.

Beyond dataset density is diversity. The datasets of [Romberg et al., 2011] and [Joly and Buisson, 2009] include the same logo under different lighting and backgrounds, while ICONS-50 has several different stylings and schemas for each class. For example, a plain class like "Cloud" has icons in which clouds cooccur with the sun, lightning, rain, and snow. And each of these different schemas include different styles and they include different renditions guided by that style (icon files partly changed across operating system versions). In summary the new ICONS-50 style robustness dataset has relatively greater dataset density and diversity.

4.2 Style Robustness

Holding out an ICONS-50 Style. The ICONS-50 dataset has several styles like Microsoft's flat vector graphics style and Apple's realistic style. We aim to test the model's robustness to an unseen style. To that end, we train a network on ICONS-50 while holding out Microsoft-styled icons. Microsoft-styled icons appear in all 50 classes, so each class is tested. Here, the metric for style robustness is simply the classifier's accuracy on Microsoft-styled icons. That said, we use a 16-4 Wide Residual Network [Zagoruyko and Komodakis, 2016] trained for 50 epochs with the cosine learning rate schedule [Loshchilov and Hutter, 2017], with a dropout [Srivastava et al., 2014] rate of 30%. Like others, we use image cropping and mirroring data augmentation. What we find is that the network lacks style robustness—the network only obtains 57% accuracy on the held out Microsoft-styled icons. This result is not symptomatic of a training data shortage, rather a lack of robustness; if we instead hold-out Apple icons (which are far more similar to other icons so do less to test style robustness), the accuracy is around 93%. With only 57% accuracy on held out Microsoft-styled icons, the network demonstrates a clear lacking in style robustness.

Style Transfer. Classifiers also have difficulty classifying ImageNet images stylized with style transfer. Style transfer [Gatys et al., 2016] provides a synthetic way to test structure robustness by algorithmically recomposing the image in the style of another image. We test robustness to style transfer in the same way we tested robustness in our IMAGENET-C corruption benchmark. Therefore we corrupt ImageNet validation images with the style transfer method of Huang and Belongie [2017]. Images are corrupted at five different severities where higher severities preserve less content and more strictly adhere to a randomly chosen target style. Improving style transfer robustness proves to be more difficult than improving IMAGENET-C performance. From AlexNet to ResNet-50, the mCE of IMAGENET-C distortions went from 100% to 76.7%, but the Corruption Error for style transfer only decreased from 100% to 92%. In fact, the ResNet-50 style transfer Corruption Error is higher than all IMAGENET-C Corruption Errors, and this holds for nearly all architectures tested. This again shows that current classifiers are not yet suited to structural changes like style.

4.3 Subclass Robustness

Telling apart a bird from a cat is simple, even if the bird species is novel. This is not seamless for classifiers, as our *subclass robustness* benchmarks show. Testing subclass robustness involves holding out a fraction of subclasses (e.g., "dove," "tabby cat") of superclasses (e.g., "bird," "cat"), training on many other subclasses for each superclass, and then testing the classifier's accuracy in predicting the superclass of the held out subclasses. Since CIFAR-100 [Krizhevsky, 2009], ImageNet-22K, and Icons-50 have taxonomies, they give us three ways to benchmark the classifier's subclass robustness. We opt to keep these benchmarks separate rather than amalgamating performance into one metric.

CIFAR-100. The CIFAR-100 dataset has 20 superclasses, each with five subclasses. For this experiment, we hold-out 1, 2, or 3 subclasses from each superclass, train the classifier on the remaining subclasses while predicting the superclass, then test the classifier's superclass accuracy on the held out subclasses. Training details are left to Appendix F. When one subclass is held out per superclass, the classifier accuracy on the held out subclasses is 40.9%. When two are held out per superclass, 26.78%; with three held out, 25.3%. Evidently there is a dearth of subclass robustness.

ImageNet-22K. Another natural image dataset with many classes but more data is ImageNet-22K, an ImageNet-1K superset. To define this subclass robustness experiment, we manually select 25 superclasses from ImageNet-22K, listed with their WordNet IDs in Appendix G. Each superclass has many subclasses. We call a subclass "seen" if and only if it is in ImageNet-1K and a subclass of one of the 25 superclasses. The subclass is "unseen" if and only if it is a subclass of the 25 superclasses and is from ImageNet-22K but not ImageNet-1K. Fortunately, pre-trained ImageNet-1K classifiers are readily available and have not trained on subclasses which should remain unseen. Therefore, we test subclass robustness by fine-tuning several pre-trained ImageNet-1K classifiers on seen subclasses so that they predict one of 25 superclasses. Their "seen" and "unseen" accuracies are shown in Figure 6, while the ImageNet-1K classification accuracy before fine-tuning is on the horizontal axis. Despite only having 25 classes, these classifiers could be far more robust to unseen subclasses.

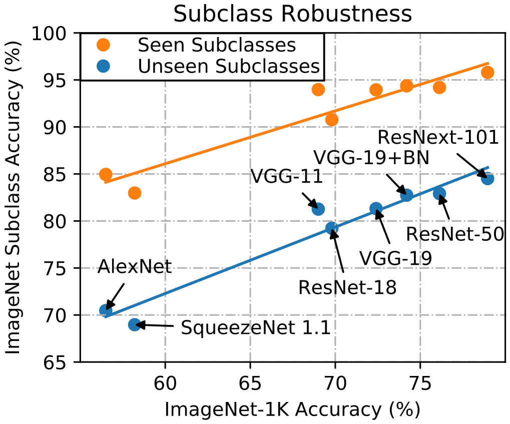

Figure 6: ImageNet classifiers and their robustness to unseen subclasses. Unseen subclasses of known superclasses are noticeably harder for classifiers.

Icons-50. Each Icons-50 class has many subclasses, so we treat each class as a superclass. Then, we can hold out 50 subclasses, train the classifier on the remaining subclasses to predict the superclass, and test the classifier's accuracy on the held out subclasses. Training details are left to Appendix F. After training, we find that classifier accuracy on these held out subclasses is a meager 60.4%. This and the previous two subclass robustness benchmarks indicate that classifiers have wide room for improvement, and together the three benchmarks give a detailed picture of subclass robustness.

5 Conclusion

In this paper, we introduced what are to our knowledge the first comprehensive benchmarks of corruption robustness and structural robustness. This was made possible by introducing two new datasets, ImageNet-C and Icons-50, the first of which showed that many years of architectural advancements corresponded to minuscule changes in relative robustness. Therefore benchmarking and improving robustness deserves attention, especially as top-1 accuracy nears its ceiling. We found that some methods harm corruption robustness, while methods such as histogram equalization, multiscale architectures, and larger models improve robustness. Afterward, we opened research in structural robustness by defining style and subclass robustness. Here we found modern models to be fragile, sometimes more fragile to structural changes than to corruptions. In this work, we had several findings, introduced novel experiments, and created new datasets for the rigorous study of model robustness, a pressing necessity as models are unleashed into the safety-critical real-world settings.

A Example of IMAGENET-C Severities

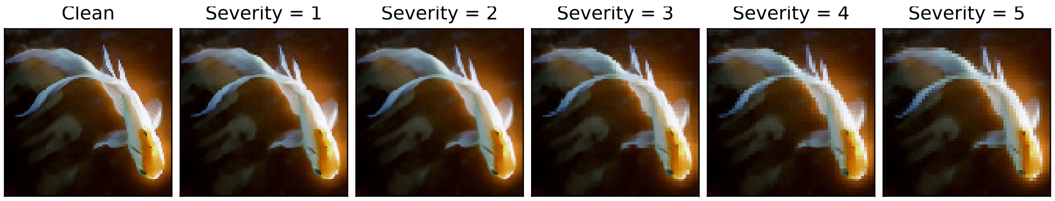

Figure 7: Pixelation modestly to markedly corrupts a fish, showcasing our benchmark's varying severities.

In Figure 7, we show the Pixelation corruption type in its five different severities. Clearly, IMAGENET-C corruptions can range from negligible to pulverizing. Because of this, the benchmark comprehensively assess each corruption type.

B Extra IMAGENET-C Corruptions

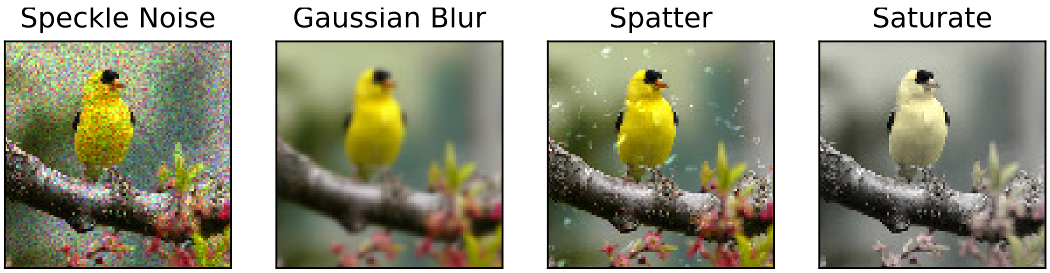

Figure 8: Extra IMAGENET-C corruption examples are available for model validation and sounder experimentation.

Directly fitting the types of IMAGENET-C corruptions is worth avoiding, as it would cause researchers to overestimate a model's robustness. Therefore, it is incumbent on us to simplify model validation. For this reason, we provide extra corruptions that are available for download [anonymized]. There is one corruption type for each noise, blur, weather, and digital category. The first corruption type is *speckle noise*, an additive noise where the noise added to a pixel tends to be larger if the original pixel intensity is larger. *Gaussian blur* is a low-pass filter where a blurred pixel is a result of a weighted average of its neighbors, and farther pixels have decreasing weight. *Spatter* can occlude a lens in the form of rain or mud. Finally, *saturate* is common in edited images where images are made more or less colorful. See Figure 8 for instances of each corruption type.

C Full Corruption Robustness Results

IMAGENET-C corruption robustness results are in Table 1 and Table 2. Since we use AlexNet errors to normalize Corruption Error values, we now specify the value $\frac{1}{5}\sum_{s=1}^{5} E_{s,\text{Corruption}}^{\text{AlexNet}}$ for each corruption type. Gaussian Noise: 88.6%, Shot Noise: 89.4%, Impulse Noise: 92.3%, Defocus Blur: 82.0%, Glass Blur: 82.6%, Motion Blur: 78.6%, Zoom Blur: 79.8%, Snow: 86.7%, Frost: 82.7%, Fog: 81.9%, Brightness: 56.5%, Contrast: 85.3%, Elastic Transformation: 64.6%, Pixelate: 71.8%, JPEG: 60.7%, Speckle Noise: 84.5%, Gaussian Blur: 78.7%, Spatter: 71.8%, Saturate: 65.8%, Style Transfer: 75.7%.

D 10-Crop Classification Fails to Enhance Robustness

Viewing an object at several different locations may give way to a more stable prediction. Having this intuition in mind, we perform 10-crop classification. 10-crop classification is executed by cropping

			Noise			Blur				Weather				Digital			
Network	Error	**mCE**	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
AlexNet	43.5	100.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
SqueezeNet	41.8	104.4	107	106	105	100	103	101	100	101	103	97	97	98	106	109	134
VGG-11	31.0	93.5	97	97	100	92	99	93	91	92	91	84	75	86	97	107	100
VGG-19	27.6	88.9	89	91	95	89	98	90	90	89	86	75	68	80	97	102	94
VGG-19+BN	25.8	81.6	82	83	88	82	94	84	86	80	78	69	61	74	94	85	83
ResNet-18	30.2	84.7	87	88	91	84	91	87	89	86	84	78	69	78	90	80	85
ResNet-50	23.9	76.7	80	82	83	75	89	78	80	78	75	66	57	71	85	77	77

Table 1: Corruption Error and mCE values of different corruptions and architectures on ImageNet-C. The mCE value is the mean Corruption Error of the corruptions in Noise, Blur, Weather, and Digital columns. All models are trained on clean ImageNet images, not ImageNet-C images. Here "BN" abbreviates Batch Normalization [Ioffe and Szegedy, 2015].

			Noise			Blur				Weather				Digital			
Network	Error	**Rel. mCE**	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
AlexNet	43.5	100.0	100	100	100	100	100	100	100	100	100	100	100	100	100	100	100
SqueezeNet	41.8	117.9	118	116	114	104	110	106	105	106	110	98	101	100	126	129	229
VGG-11	31.0	123.3	122	121	125	116	129	121	115	114	113	99	86	102	151	161	174
VGG-19	27.6	122.9	114	117	122	118	136	123	122	114	111	88	82	98	165	161	172
VGG-19+BN	25.8	111.1	104	105	114	108	132	114	119	102	100	79	68	89	165	125	144
ResNet-18	30.2	103.9	104	106	111	100	116	108	112	103	101	89	67	87	133	97	126
ResNet-50	23.9	105.0	104	107	107	97	126	107	110	101	97	79	62	89	146	111	132

Table 2: Relative Corruption Errors and Relative mCE values of different corruptions and architectures on ImageNet-C. All models are trained on clean ImageNet images, not ImageNet-C images. Here "BN" abbreviates Batch Normalization.

all four corners and cropping the center of an image. These crops and their horizontal mirrors are processed through a network to produce 10 predicted class probability distributions. We average these distributions to compute the final prediction. Of course, a prediction informed by 10-crops rather than a single central crop is more accurate. Ideally, this revised prediction should be more robust too. However, the gains in mCE do not outpace the gains in accuracy on a ResNet-50. In all, 10-crop classification is a computationally expensive option which contributes to classification accuracy but not noticeably to robustness.

E Classes in Icons-50

The 50 classes of Icons-50 are as follows: Airplane, Arrow Directions, Ball, Biking, Bird, Blade, Boat, Books, Building, Bunny Ears, Cartwheeling, Clock, Cloud, Disk, Drink, Emotion Face, Envelope, Family, Fast Train, Feline, Flag, Flower, Footwear, Golfing, Hand, Hat, Heart, Holding Hands, Japanese Ideograph, Kiss, Lock, Mailbox, Marine Animal, Medal, Money, Monkey, Moon, Mountain, Numbers, Phone, Prohibit Sign, Star, Surfing, Tree, Umbrella, Vehicle, Water Polo, Worker, Wrestling, Writing Utensil.

F CIFAR-100 and Icons-50 Subclass Robustness Training Setup

The CIFAR-100 superclass classifier is a 40-2 Wide ResNet. The network trains for 100 epochs with a dropout rate of 0.3. The initial learning rate of 0.1 decays following a cosine learning rate schedule. We use standard flipping and data croppting augmentation. The batch size is 128, and the optimizer is Stochastic Gradient Descent with Nesterov momentum.

Separately, the Icons-50 classifier is a 16-4 Wide ResNet. Other than the architecture, the primary differences from the CIFAR-100 training setup are that we train for 50 epochs and that we need to resize the icon images to 32 × 32 images, whereas in CIFAR-100 they are already 32 × 32.

G Selected ImageNet-22K Superclasses

The 25 superclasses that we select from ImageNet are as follows: Amphibian (n01627424), Appliance (n02729837), Aquatic Mammal (n02062017), Bird (n01503061), Bear (n02131653), Beverage (n07881800), Big cat (n02127808), Building (n02913152), Cat (n02121620), Clothing (n03051540), Dog (n02084071), Electronic Equipment (n03278248), Fish (n02512053), Footwear (n03380867), Fruit (n13134947), Fungus (n12992868), Geological Formation (n09287968), Hoofed Animal (n02370806), Insect (n02159955), Musical Instrument (n03800933), Primate (n02469914), Reptile (n01661091), Utensil (n04516672), Vegetable (n07707451), Vehicle (n04576211).

References

Ossama Abdel-Hamid, Abdel rahman Mohamed, Hui Jiang, and Gerald Penn. Applying convolutional neural networks concepts to hybrid nn-hmm model for speech recognition. *ICASSP 2013*. 2

Ángel de la Torre, Antonio Peinado, José Segura, José Pérez-Córdoba, Ma Carmen Benítez, and Antonio Rubio. Histogram equalization of speech representation for robust speech recognition. *IEEE Signal Processing Society*, 2005. 2, 5

Osbert Bastani, Yani Ioannou, Leonidas Lampropoulos, Dimitrios Vytiniotis, Aditya Nori, and Antonio Criminisi. Measuring neural net robustness with constraints. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2613–2621. Curran Associates, Inc., 2016. 2

Antoni Buades and Bartomeu Coll. A non-local algorithm for image denoising. In *CVPR 2005*, 2005. 5

Nicholas Carlini and David Wagner. Defensive distillation is not robust to adversarial examples, 2016a. 1, 2

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks, 2016b. 2

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods, 2017. 1, 2

Nicholas Carlini, Guy Katz, Clark Barrett, and David L. Dill. Ground-truth adversarial examples, 2017. 2

Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *CVPR 2009*, 2009. 2

Samuel Dodge and Lina Karam. Quality resilient deep neural networks, 2017a. 2

Samuel Dodge and Lina Karam. A study and comparison of human and deep learning recognition performance under visual distortions, 2017b. 2

David Donoho and Iain Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 1993. 5

Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. Robust physical-world attacks on deep learning models, 2017. 2

Leon Gatys, Alexander Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. *CVPR 2016*, 2016. 7

Robert Geirhos, David H. J. Janssen, Heiko H. Schütt, Jonas Rauber, Matthias Bethge, and Felix A. Wichmann. Comparing deep neural networks against humans: object recognition when the signal gets weaker, 2017. 2

Mark Harvilla and Richard Stern. Histogram-based subband powerwarping and spectral averaging for robust speech recognition under matched and multistyle training, 2012. 2, 5

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CVPR 2016*, 2015. 4

Dan Hendrycks and Kevin Gimpel. Early methods for detecting adversarial images, 2016. 2

Hans-Günter Hirsch. Aurora-5 experimental framework for the performance evaluation of speech recognition in case of a hands-free speech input in noisy environments, 2007. 2

Hans-Günter Hirsch and David Pearce. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. *ISCA ITRW ASR2000*, 2000. 2

Hossein Hosseini, Baicen Xiao, and Radha Poovendran. Google's cloud vision api is not robust to noise, 2017. 2

Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q. Weinberger. Multi-scale dense networks for resource efficient image classification. *ICLR 2018*, 2017a. 5

Gao Huang, Shichen Liu, Laurens van der Maaten, and Kilian Q Weinberger. Condensenet: An efficient densenet using learned group convolutions. *arXiv preprint*, 2017b. 5

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017c. 5

Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. *ICCV*, 2017. 7

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *JMLR 2015*, 2015. 10

Alexis Joly and Olivier Buisson. Logo retrieval with a contrario visual query expansion. *ACM International Conference on Multimedia Retrieval 2009*, 2009. 6, 7

Guy Katz, Clark Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Towards proving the adversarial robustness of deep neural networks. *arXiv 2017*, 2017. doi: 10.4204/EPTCS.257.3. 2

Tsung-Wei Ke, Michael Maire, and Stella X. Yu. Multigrid neural architectures, 2016. 5

Chanwoo Kim and Richard M. Stern. Power-normalized cepstral coefficients (pncc) for robust speech recognition. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.*, 24(7):1315–1329, July 2016. ISSN 2329-9290. 2

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 8

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NIPS 2012*, 2012. 1, 4

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *ICLR 2017*, 2017. 2

Jinyu Li, Li Deng, Yifan Gong, and Reinhold Haeb-Umbach. An overview of noise-robust automatic speech recognition. volume 22, pages 745 – 777. IEEE – Institute of Electrical and Electronics Engineers, April 2014. 2

Fu-Hua Liu, Richard M. Stern, Xuedong Huang, and Alex Acero. Efficient cepstral normalization for robust speech recognition. In *Proc. of DARPA Speech and Natural Language Workshop*, pages 69–74, March 1993. 2

Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2017. 7

Jiajun Lu, Hussein Sibai, Evan Fabry, and David Forsyth. Standard detectors aren't (currently) fooled by physical adversarial stop signs, 2017. 2

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2017. 2

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations, 2017. 2

Vikramjit Mitra, Horacio Franco, Richard Stern, Julien Van Hout, Luciana Ferrer, Martin Graciarena, Wen Wang, Dimitra Vergyri, Abeer Alwan, and John H.L. Hansen. Robust features in deep learning based speech recognition, 2017. 2

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks, 2017. 2

Stephen M. Pizer, E. Philip Amburn, John D. Austin, Robert Cromartie, Ari Geselowitz, Trey Greer, Bart Ter Haar Romeny, and John B. Zimmerman. Adaptive histogram equalization and its variations. *Computer Vision, Graphics, and Image Processing*, 1987. 5

Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox v0.8.0: A python toolbox to benchmark the robustness of machine learning models, 2017. 2

Stefan Romberg, Lluis Pueyo, Rainer Lienhart, and Roelof van Zwol. Scalable logo recognition in real-world images. *ACM International Conference on Multimedia Retrieval 2011*, 2011. 6, 7

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *JMLR 2014*, 2014. 7

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2013. 1, 2

Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks, 2016. 2

Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. *CVPR 2017*, 2016. 5

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 7

Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training, 2016. 2, 4