THE UNIVERSITY OF CHICAGO

TOWARDS ROBUSTNESS OF NEURAL NETWORKS

A DISSERTATION SUBMITTED TO
THE FACULTY OF THE DIVISION OF THE PHYSICAL SCIENCES
IN CANDIDACY FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

DEPARTMENT OF COMPUTER SCIENCE

BY
STEVEN BASART

CHICAGO, ILLINOIS
DECEMBER 2021

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I want to thank my parents who have always supported me. I owe this day to them. I love you mom and pop. I want to thank all of my professors, especially the professors who went above and beyond as mentors. Thank you Greg for helping me beyond researching ideas to also thinking about life, and what I will do beyond schooling. Thank all of my friends and colleagues, Dan, Mohammadreza, Mantas, Gustav, and Payman all of whom in one way or another helped me achieve this goal. Also would like to thank my friends Matt and Kevin. I would also like to thank Keziah for helping me edit the thesis. I would like to thank Victoria for also helping with edits, and being by my side. Finally I would like to thank everyone who have helped to make this day possible.

ABSTRACT

We introduce several new datasets namely ImageNet-A/O and ImageNet-R as well as a synthetic environment and testing suite we called CAOS. ImageNet-A/O allow researchers to focus in on the blind spots remaining in ImageNet. ImageNet-R was specifically created with the intention of tracking robust representation as the representations are no longer simply natural but include artistic, and other renditions. The CAOS suite is built off of CARLA simulator which allows for the inclusion of anomalous objects and can create reproducible synthetic environment and scenes for testing robustness. All of the datasets were created for testing robustness and measuring progress in robustness. The datasets have been used in various other works to measure their own progress in robustness and allowing for tangential progress that does not focus exclusively on natural accuracy.

Given these datasets, we created several novel methods that aim to advance robustness research. We build off of simple baselines in the form of Maximum Logit, and Typicality Score as well as create a novel data augmentation method in the form of DeepAugment that improves on the aforementioned benchmarks. Maximum Logit considers the logit values instead of the values after the softmax operation, while a small change produces noticeable improvements. The Typicality Score compares the output distribution to a posterior distribution over classes. We show that this improves performance over the baseline in all but the segmentation task. Speculating that perhaps at the pixel level the semantic information of a pixel is less meaningful than that of class level information. Finally the new augmentation technique of DeepAugment utilizes neural networks to create augmentations on images that are radically different than the traditional geometric and camera based transformations used previously. DeepAugment improves SOTA by a significant margin while being able to be used with other augmentation schemes and demonstrates that neural augmentations are not only possible but provide a benefit with respect to robustness.

CHAPTER 1

INTRODUCTION

Machine Learning (ML) models are becoming more widespread in their use and adoption. As their use becomes more prevalent in safety-critical applications or trust necessary situations, the models must exhibit reliability to ensure their continued adoption. Some of the current domains that exhibit these properties include self-driving vehicles and in medical diagnoses; where in self-driving, both trust and safety are paramount. In the second domain of medical diagnoses trust is the primary factor such that a model will need to be able to provide explanations or a confidence such that other professionals can best decide how to proceed with the outputted information.

Robustness is of especial importance in machine learning where many of the commonly used models are uncalibrated out of the box. All of the works thus far that aimed at addressing this short-coming also come at the cost of model accuracy. While calibration is of importance for building confidence in model predictions, it only represents one facet of robustness. In this thesis we will cover several aspects of robustness and begin to show that robustness covers several related areas. While we explore the related areas, we also aim to highlight the delineations between them. These delineations help researchers better focus on specific problem domains, which then allows the community to make improvements. Beyond covering the delineations, we also demonstrate that there are still unknown areas of robustness that might not have proper categories or delineations at the moment.

We define *robustness* as the preservation of functionality under changing conditions. Within the ML community robustness has grown to encompass several different concepts which include the following domains: generalization, sensitivity, distribution shift, adversarial examples, out-of-distribution (OOD) detection, calibration, and even interpretability. We shall attempt to make those distinctions clear. We give a detailed explanations of the domains in Section 2.1 and a brief description of the OOD detection task in Section 2.4. This task fits into robustness by trying to preserve confidence on in-distribution (ID) examples in the presence of OOD examples. Therefore, the OOD detection task is useful for building safe ML systems. For example in robotics systems it

can serve as an indicator for the robot to hand off to an operator and in medicine, it can signal that the input is malformed and should be redone or handled by a doctor.

We organize the thesis loosely based around machine learning tasks covering multi-class classification first, then segmentation, then multi-label, and finally meta-learning. Segmentation can be viewed as a multi-class classification problem applied to every individual pixel. Multi-label classification can be thought of as an extension of multi-class classification so we cover multi-label after multi-class and segmentation. This extension is not perfect and we show some limitations that comes from trying to generalize successes from multi-class to multi-label. We then discuss meta-learning with respect to robustness which is distinct enough from multi-class and multi-label to be covered last.

1.1 Contributions

Our contributions to robustness research are as follows:

- Datasets: In this thesis we contribute three distinct datasets to help push the research area of robustness further. The first dataset explores the current errors that models make and are termed hard-negative examples or natural adversarial examples in Chapter 3. The second dataset involves the creation of a controllable anomalous segmentation dataset created by modifying the CARLA simulator in Chapter 4. Finally, the third dataset aims to better explore distribution shift from ImageNet-1K to other representations such as cartoons, sculptures, or tattoos among others in Chapter 5.

- New Robustness Techniques: We utilize the Kullback–Leibler divergence (KL-Divergence) to improve OOD detection in the multi-class setting. We further show that using the maximum logit serves as a better indicator of OOD detection than the previous baseline. Both of those results are presented in Chapter 4. We also develop a novel non-linear augmentation technique in Chapter 5. The novel technique is able to greatly improve robustness in the multi-class setting and highlights a new unexplored area that can also help generalization.

- Novel Field Criticisms: By examining different variations of robustness, we are able to observe where current techniques are still lacking in Chapters 5 and 6. Most notably is the recognition that almost all of the previous robustness techniques failed to generalize to the multi-label setting.

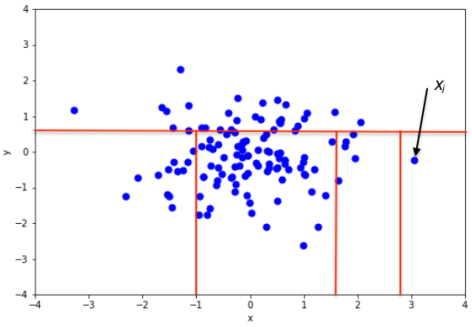
- Few shot robustness: We explore the relationship of robustness and few shot learning in Chapter 7. By converting the metric learning task into a set-membership task we find that while the models are able to learn set-membership for in-distribution classes, the models are unable to generalize to novel unseen classes.

CHAPTER 2

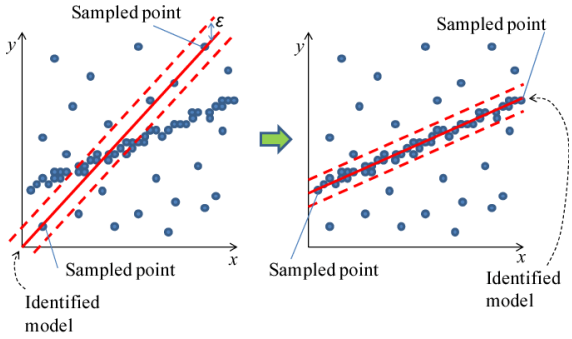
BACKGROUND

2.1 Robustness

We shall cover a few seemingly disparate robustness topics and show how they relate together. In this thesis, we focus on natural out-of-distribution (OOD) detection. Since ML systems are deployed in natural settings, natural OOD detection is of utmost importance and should be solved immediately. In addition to natural OOD detection, in Chapter 4, we explore anomaly segmentation in simulated and adversarial examples. We explore these areas to study whether any of the techniques in simulations or improvements in adversarial robustness generalize to natural settings. In Section 2.2 we cover adversarial examples in greater detail.



(a) Points are drawn from a 2D Gaussian, and the isolation forest model is labeling the point X as an outlier.



(b) Given differently sampled points leads to different lines which causes some number of points to be labeled as outliers. After sampling sufficient number of points then the best line is determined. Image credit: Watanabe 2013.

Figure 2.1: Examples of two novelty detection methods.

OOD detection has its roots in outlier and novelty detection. The main distinction between the more recent OOD detection and outlier detection are the assumptions on what is considered an outlier. In OOD detection, the task is set up similarly to a classification problem by having an in-distribution set, and an unknown and much larger out-of-distribution set. Whereas in outlier detection the assumption would be that there exists a small subset of data which are considered outliers, and hence the data provided is considered "poisoned." The commonly used approach in

statistics is to first define the model and identify the outliers as points with low probability mass. More advanced methods exist such as using local outlier factor (M. Breunig et al. 2000) or using RANSAC (Fischler and Bolles 1981) to determine the inliers versus outliers. See figure 2.1 for a visual demonstration of the techniques. The former method works by comparing the distances of points to their neighbors and finding a threshold for the computed distance. However many of these methods still rely on determining the model first to fit the data.

In a setting closely related to outlier detection, there is the problem of training with known label corruptions. Both domains deal with the case of data poisoning, while this sub-problem exclusively concerns itself with poison occurring in the labels. Furthermore the problem of label corruption can be approached from the assumption whether there exists a subset of trusted labels. Charikar, Steinhardt, and Valiant 2017 analyze both conditions to give theoretical guarantees and an algorithm to use under each setting. Patrini et al. 2017 address the problem in multi-class classification with no trusted data where they first estimate the level of corruption, then given the confusion matrix apply it after the network's outputs to correct for the uncertainty. Ren et al. 2018 also approach the problem under the same assumptions, but they use meta-learning (see Section 2.3) to assign and reweight the examples used for training. Follow up work has shown that MixUp is also an effective way to increase the training given label corruption (Arazo et al. 2019). While the problem remains unsolved, some works highlight that for some models, namely neural networks, label noise might not be as much of a problem compared to other ML methods as the models tend to be robust to out of the box (Rolnick et al. 2017).

On the other hand, novelty detection, unlike in outlier detection, operates under the assumption that the dataset is not polluted with outliers, and instead tries to detect if a new example is an outlier. Some classical techniques in novelty detection are that of isolation forest (Liu, Ting, and Zhou 2008) and one-class SVM (Schölkopf et al. 1999a). These methods employ a training and testing phase. Some of the methods can not give estimations of outliers for the training set by construction. This set-up more closely reflects the problems in OOD detection. However, many of the novelty detection methods such as those listed exhibit poor performance (A. F. Emmott

et al. 2013).

In contrast to outlier and novelty detection, OOD detection frames the problem of detecting anomalies as a learning problem. Novelty detection can be considered to be within unsupervised classification, while OOD detection expands on novelty detection by considering supervised and semi-supervised along with unsupervised classification. Hendrycks and Gimpel 2017 uses a trained neural network's output as a confidence score for detecting anomalies in the setting of multi-class classification. As a followup work, ODIN (Liang, Li, and Srikant 2018) and Mahalanobis detector (Lee et al. 2018b) apply small gradient perturbations derived from label corruptions and then run the corrupted image through the network again to get the final output score. The work by Lee et al. 2018b modifies this approach by training a per-class dependent model on the perturbations. The main issue with the previous two approaches in particular is that they fine-tune their corruptions on the different perturbation types which leads to a form of training on the test set. Other methods such as the confidence estimator trains a separate confidence branch to directly predict both the class label and the confidence of the prediction (DeVries and Taylor 2018). In Chapter 5 we further explore the relationship between data augmentation and robustness by demonstrating how random corruptions by a neural network improve OOD detection.

2.2 Adversarial Examples

Adversarial examples are modifications to inputs, such as images, that causes the label of a function (typically a deep neural network) to change, while a human would still classify the input as belonging to the same class. It was first discovered that neural networks are susceptible to these small perturbations in Szegedy et al. 2014. Figure 2.2 shows how small adversarial noise can affect the outputs of a deep neural network (DNN). This work spawned several works in interpretability such as Olah, Mordvintsev, and Schubert 2017 and Selvaraju, Cogswell, et al. 2019 whereby the gradient perturbations were used to visualize what the network was attending to. Later on after (Carlini and Wagner 2017)'s research findings, the research community began to focus on the problem treating it as an actual security threat threat instead of as an anomaly.

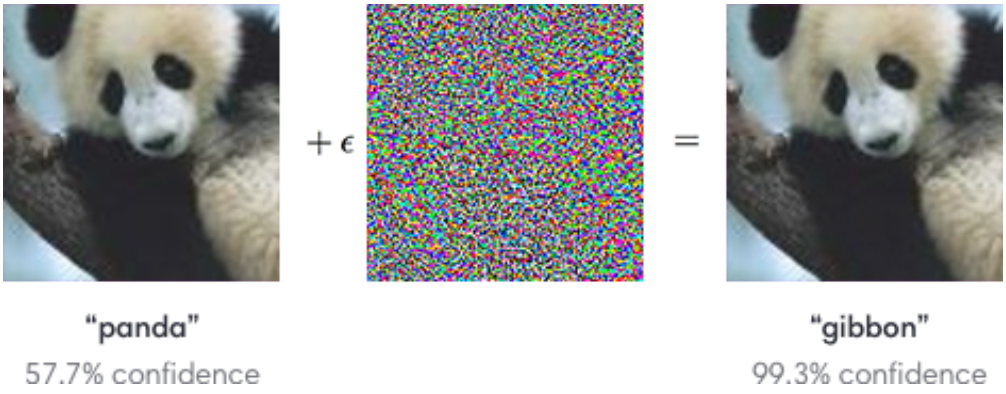


Figure 2.2: An adversarial example showing how a small perturbation can change the label of an image. The adversarial noise added to image is magnified by 1000x to be visible to a human. Image credit: OpenAI

Let us formally define adversarial examples. Note for the purposes of these definitions we will restrict the domain of examples or inputs to be that of images but it can extended to other domains such as speech or language. An input image is $\mathbf{x}$ and a trained model is f (typically a neural network). $f(\cdot)$ is the output probability distribution for a set of classes the model was trained on. We will perturb $\mathbf{x}$ by a small amount ϵ which will result in our adversarial example $\mathbf{x}' := \mathbf{x} + \epsilon$. We say $\mathbf{x}'$ is an adversarial example if $\arg\max_{\mathbf{x}} f(\mathbf{x}) \neq \arg\max_{\mathbf{x}'} f(\mathbf{x}')$, it is also described as a successful attack. ϵ is of the same dimensionality as $\mathbf{x}$ but is bounded by some distance metric. We use several L_p distance measures to bound ϵ as a proxy for the measure we actually care for, which is human perceptual distance. More recently, there is work trying to move beyond ℓ_p distances such as (Bhattad et al. 2020) because bounding $\|\varepsilon\|_\infty \leq \delta$ distance is a somewhat arbitrary limitation on the attacker.

It is common practice in the security research community to detail what the assumptions are concerning the threat model, which includes what the attacker has access to, capabilities, and limitations. Most of what has so far been discussed relates to the scenario where the attacker has full access to the machine and the model such that they can inject signal noise ϵ into the system to be able to change the resulting classification output of f. While this setting has received much attention, it is largely unrealistic as an actual security threat. More recently, there have been efforts to change the threat model to consider more real world attacks such as (Eykholt et al. 2018;

Ian J. Goodfellow, Shlens, and Szegedy 2015; Yakura and Sakuma 2019) where the attacker has a more limited access to the model via its external sensors.

As an implicit goal in desiring to make the models more similar to humans, we treat adversarial examples as inherent flaws in the model. There are some discussions concerning whether or not adversarial examples are unavoidable (Engstrom et al. 2019), although this view is currently held by a minority in the research community. We shall detail a few of the current hypotheses being investigated for why adversarial examples might arise. One of the hypotheses by Ian J. Goodfellow, Shlens, and Szegedy 2015 purports that the brittleness is a result of excessive linearity. The high-dimensionality of the inputs allows an adversary easy access to cross a decision boundary of a class. The simple example is for a two class problem given a powerful enough adversary, the inputs can be moved along any direction and with high probability half of the directions will cross the decision boundary. This hypothesis has been indirectly challenged by subsequent work (Wong and Kolter 2018). Another hypothesis is that of robust and non-robust features (Ilyas et al. 2019). The argument is that the networks are picking up on highly predictive features that do well on the training set, but are too brittle and incomprehensible for humans which then allows for adversarial examples. Some qualitative examples supporting this claim are that adversarially trained models have features that are more recognizable to humans. The features themselves are trained on limited datasets which creates an artificial selection bias and lacks more theoretical underpinnings. There is no definitive consensus for explanations or solutions to the problem of adversarial examples at the current time.

While there does not exist a solution to the problem, there are several defenses that can mitigate the power of adversarial attacks. When a defense is "broken" it refers to the attacker reducing the accuracy of the model to zero (or nearly zero). The most prominent defenses for adversarial examples are some form of training with adversarial examples. The original formulation from A. Madry et al. 2018 adds an adversarial example during training as extra training data for the model to learn from. Followup work from Hongyang Zhang et al. 2019 explicitly considers the tradeoff between natural and adversarial accuracy by incorporating an additional loss term for

the adversarial examples used during training. There are other works that provide certifications for limited threat models such as L_2 distance (Cohen, Rosenfeld, and Kolter 2019) and other forms of certifications are being explored. Methods which provide some amount of robustness include the following: Compression of either inputs, models, or both (Liu et al. 2019; Dziedzic et al. 2019); injecting noise into either the input examples, models, or both (You et al. 2019; Cohen, Rosenfeld, and Kolter 2019); data augmentation, or pre-training (Hendrycks, Lee, and Mazeika 2019); and ensembling which use multiple models to give one output using a voting, or some other scheme (Tramèr et al. 2018).

Part of the difficulty with comparing defenses is the lack of standardization to compare the different threat models and environments. Recently, projected gradient descent attacks with ℓ_∞ perturbation of size 8/255 has become the standard because greater perturbations allowed for changing the label to a human annotator (Tramèr, Behrmann, et al. 2020). Given this limited adversary/scenario allows for direct comparison between methods. The other issue is that the threat model or environment is being an unfair or being an unreasonable limitation to the adversary. Due to this limitation researchers use adative attacks (Tramèr, Carlini, et al. 2020) which allow the attacker to know the defense and modify the attack given this knowledge. Given the relative infancy of the joint domains of ML and computer security, the field will eventually settle on realistic threat models and the issue of defense comparison will become moot.

Although the focus is mostly on neural networks, other classical machine learning (ML) models are also susceptible to adversarial noise such as k Nearest Neighbors (k-NN), SVMs, and linear regression. These other machine learning models have different failure modes for adversarial examples but have all been susceptible to the attacks. Gilmer et al. 2019 suggests that the previous methods fail due to the high dimensionality of the inputs. Currently neural networks provide the best defenses against adversarial examples (Goodfellow 2016). However, the other ML models have been less explored both to attack and defend against this threat model.

It is worth noting the difficulty of transitioning to real-world attacks via adversarial examples. Bhattad et al. 2020 demonstrates that artificial constraints had to be placed on the model for the

adversary to succeed. This is not to suggest that the signal based attacks are unimportant, but it should be studied to determine how well adversarial attacks can survive the transition to complex environments, to 3D, and against multiple modalities. Dosovitskiy et al. 2017 have shown how difficult it is to construct adversarial examples in 3D environments.

2.3 Meta-Learning

Meta-learning is most commonly understood as the task of "learning to learn", which refers to the process of improving a learning algorithm over distributions. This contrasts with the conventional machine learning, which is the process of learning a model for a single distribution or over many data instances. Meta-learning is comprised of a two stage process. The first stage involves a base learner and the second stage (also confusingly called meta-learning) involves an outer algorithm that optimizes an outer objective. An example of meta-learning the inner (or lower, base) learning algorithm solves a task such as image classification (Franceschi et al. 2018), defined by a dataset and objective. Then during meta-learning, an outer (or upper, meta) algorithm updates the inner one, so that the learned model can perform well in few-shot learning (Hospedales et al. 2020).

Recent work has tried to minimize the distinctions between meta-learning and supervised classification (Chao et al. 2020). Maurer 2005 extended the generalization error bounds from supervised learning to meta-learning. Formulating the problem of meta-learning more closely with supervised learning better fits the theme of this thesis trying to improve OOD detection.

We most prominently explore two techniques in meta-learning. The first technique is from Finn, Abbeel, and Levine 2017 which uses the optimization algorithm to perform the meta-learning. The inner learner does standard supervised learning for the task, e.g. image classification, and the outer optimization algorithm does an averaged gradient step for several tasks (different image classification tasks). The second technique imbues the space of learned features with a metric to be later used for a nearest neighbor classification. We ignore the other type of meta-learning which uses memory augmented networks to perform the meta-learning. Those models are currently more akin to a novelty as opposed to offering a more utilized method such as the others described

before it.

We explore meta-learning in Chapter 7 to explore how the techniques from meta-learning namely the scenario of learning in the few shot setting can better assist in generalizing to unseen OOD examples. In both scenarios the amount of OOD data far exceeds the amount of training data that the model is expected to generalize from.

2.4 Tasks

In this section, we will cover three computer vision tasks that will be addressed in this paper namely multi-class classification, multi-label classification, and segmentation,.

2.4.1 Multi-class Classification

Multi-class classification is a rich sub-field in machine learning with its origins in binary classification (Cox 1958). From binary classification there are three approaches that have been taken to handle the multi-class setting. The first approach is by reduction of the multi-class problem to binary classification, typically One-vs-Rest where one would use a classifier to distinguish one class from all of the others. The size of these approaches scale linearly with the number of classes due to requiring a different model per class. The second approach handles the problem by extending the binary classification to multi-class classification directly. Finally, the last approach handles the problems by creating hierarchies for classification (Silla and Freitas 2010).

We will be covering the second approach for multi-class classification, the extension from binary, which include methods such as k-nearest-neighbors (Altman 1992), logistic regression (Menard 2002), neural networks (Hopfield 1988), and random forests (Breiman 2004). We will briefly cover neural networks as they are utilized throughout the paper and are used in the other tasks as well. For neural networks we mostly utilize convolutional neural networks (CNN). CNNs is composed of the following operations: learned filters which perform cross-correlation, convolution, non-linearity, and pooling which reduces the dimensionality of the inputs and

features. These operations are repeated in succession to form what are considered layers before finally applying a differentiable loss function so that the network can learn via stochastic gradient descent. The specific implementation details of each neural network architecture is left to the respective chapter.

The problem in multi-class classification is to associate each instance with a unique label from a set of labels. More formally given a set of training data $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}, \forall i \in \{1, ..., N\}$ where $\mathbf{x} \in \mathbb{R}^d$, the goal is to learn (or induce) a model $f(\mathbf{x}) = \hat{\mathbf{y}}$ such that $\hat{\mathbf{y}}$ minimizes a loss function $L : \mathcal{R} \times \mathcal{R} \rightarrow \mathcal{R}$. During classification, the assumption is training and test data points are drawn i.i.d. (independent and identically distributed) from a full joint distribution. Although in most cases, the full joint distribution is unknown, a finite number of examples can be used to measure generalization by splitting the data into training and test sets. Generative and discriminative techniques are useful for learning model f. Generative techniques learn the full joint probability distribution while the latter techniques model the class or decision boundaries. In this thesis, we focus on discriminative methods for classification.

Some influential datasets in the area of multi-class classification have been MNIST (Lecun et al. 1998), SVHN (I. J. Goodfellow et al. 2013), CIFAR (Krizhevsky and Hinton 2009), and ImageNet (Russakovsky et al. 2014). MNIST and SVHN are both datasets consisting of the digits 0-9. MNIST are black and white digits, and SVHN are numbers occurring on houses which are colored images and have a variety of backgrounds. On the other hand, CIFAR and ImageNet are natural images consisting of animals, automotive vehicles, and many other categories. The former are small images with ten categories and the latter are large images with one thousand categories. These datasets, among others, have allowed for great advances in the task of multi-class classification (Ranzato et al. 2006; Beygelzimer, Kakade, and Langford 2006; Cho and Saul 2009; Alex Krizhevsky, Sutskever, and Geoffrey E Hinton 2012; Carlini and Wagner 2017).

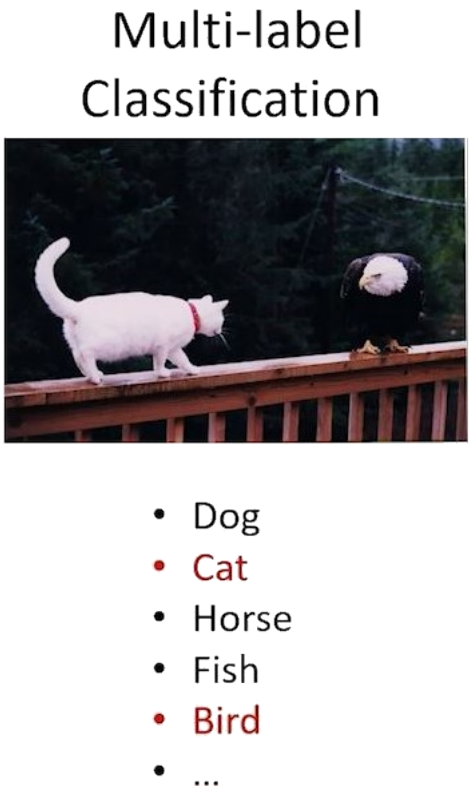


Figure 2.3: An example of an image with both a cat and a bird which shows multiple labels occurring together.

2.4.2 Multi-label Classification

Building from multi-class classification, there is multi-label classification whereby instead of the traditional setting where an instance may be of only one label, in this setting each instance is associated with a set of labels. As an example consider the figure 2.3 where it can be classified as a bird, bald eagle, cat, porch and forest. This task being a natural extension of multi-class is therefore a harder task because of determining the presence or absence of for all classes as opposed to the problem of selecting the most likely candidate. Multi-label classification technique is useful for (1) text classification involving lots of documents about several topics, (2) audio classification, where some songs are a mixture of styles or genres and (3) image search where different images or videos can capture multiple genres or styles at the same time (Brhanie 2016).

We will similarly define the task of multi-label classification. The problem in multi-label (or sometimes referred to as multi-category or multi-topic) classification is to associate each

instance with a set of labels. This again follows the same conventions and notations as multi-class classification whereby training data $D = \{(\mathbf{x}_i, \mathbf{y}_i)\}, \forall i \in \{1, ..., N\}$ contain examples x that are a d-dimensional vector. However, unlike in multi-class setting y is no longer a one-hot vector and are instead binary strings without the limitation of only one "1" in the string.

Multi-label classification is arguably better for studying robustness. The main reasons for this is because in multi-class classification the models are either forced to pick one out of the set of labels or add in a catch-all "other" class. In the first scenario the model "should" output a uniform probability distribution over all classes, however, this also seems unrealistic as there will exist OOD examples which will appear to be similar to in-distribution classes and so a uniform output appears to be an unfeasible goal. The second option seems more promising but still has a similar issue. Classifying images or parts of image as background will lead to the model "correctly" classifying OOD examples as being background. This scenario again presents issues that there is no longer any distinction between in and out-of-distribution examples, which can be a safety hazard in some scenarios. Multi-label classification does not have these two issues because having a uniform output of no class is a feasible solution that is also realistic. The secondary issue is still present unless modifications are made, that of being able to discern between in- and out-of-distribution examples.

Unfortunately, multi-label classification has not received as much attention within computer vision as that of multi-class classification. This can be observed by the historical lag in multi-label datasets from Corel-5k (Duygulu et al. 2002), then PASCAL VOC (Everingham et al. 2009) and more recently we have MSCOCO (Lin et al. 2014) and Tencent ML-image Wu et al. 2019. Similarly multi-label OOD is also a wholly underexplored area. We hope to change this by releasing some baselines for this task in Chapter 6.

2.4.3 Segmentation

Segmentation refers to the task of partitioning the pixels of an image into regions or segments, where the pixels in each region share similar attributes. The goal of segmentation is to simplify

Semantic Segmentation Instance Segmentation

Figure 2.4: Highlighting the distinction between semantic segmentation on the left and instance segmentation on the right. Image source: Arnab et al. 2018

and/or change the representation of an image into something meaningful and easier to analyze (Shapiro and Stockman 2001). We categorize segmentation methods into two main categories that of semantic and class agnostic segmentation. Class agnostic segmentation is the task of segmenting an image into the region boundaries such as foreground and background. Semantic segmentation on the other hand, deals with the task of assigning each pixel to an element of a set of classes. As a related task to semantic segmentation there is instance segmentation where the goal is to identify all distinct occurrences of the segmented objects in an image.

Class agonostic segmentation involves partitioning an image into coherent regions. Martin et al. 2001 show in Berkeley Segmentation Dataset (BSDS) that human annotations of edges and boundaries are nonrandom and useful for supervised class agnostic segmentation. However, the BSDS dataset has been most useful in unsupervised segmentation (Arbelaez et al. 2010; Martin, Fowlkes, and Malik 2004; Achanta et al. 2010; Carreira and Sminchisescu 2010), in such tasks as foreground/background segmentation (Wu and Wang 2008). Unsupervised segmentation typically involve clustering (Martin, Fowlkes, and Malik 2004) and are trained by learning a distance between pixels within segments. Their outputs have been used in other downstream tasks due to the reduction in the input space (Mostajabi, Yadollahpour, and Shakhnarovich 2015). Due to the hardware and algorithmic improvements, reliance on "super pixels" (Achanta et al. 2010) has waned and the current approaches label each pixel directly in a given image.

Unlike class agnostic segmentation, semantic segmentation is a supervised task where the goal

is to assign every pixel in an image a label from a given set of classes. In semantic segmentation multiple instances of the same object class are not differentiated. While in instance segmentation the goal is to assign a unique label to every object instance in an image. A limitation of instance segmentation is with dealing with uncountable classes such as pixels belonging to the sky or sand. The union of both semantic and instance segmentation is known as panoptic segmentation (Kirillov, He, et al. 2019). The goal of panoptic segmentation is to assign both a category label to every pixel and if the category is an instance based category, such as people, then also assign it a unique id.

Prominent segmentation datasets include BSDS (Martin et al. 2001), PASCAL-VOC (Everingham et al. 2009) and MS COCO (Lin et al. 2014). These datasets have allowed for significant progress in unsupervised and supervised segmentation (Arbelaez et al. 2010; Agrawal, Girshick, and Malik 2014; Wu and Wang 2008; Arnab et al. 2018; Kirillov, Wu, et al. 2019).

CHAPTER 3

NATURAL ADVERSARIAL EXAMPLES

3.1 Overview

Research on the ImageNet (Russakovsky et al. 2014) benchmark has led to numerous advances in classification (A. Krizhevsky, Sutskever, and Geoffrey E. Hinton 2017), object detection (Huang, Rathod, et al. 2017), and segmentation (He et al. 2017). ImageNet classification improvements are broadly applicable and highly predictive of improvements on many tasks (Kornblith, Shlens, and Le 2018). Improvements on ImageNet classification have been so great that some call ImageNet classifiers "superhuman" (He, Zhang, and Ren 2015). However, performance is decidedly subhuman when the test distribution does not match the training distribution (Hendrycks and Dietterich 2019). The distribution seen at test-time can include inclement weather conditions and obscured objects, and it can also include objects that are anomalous.

Recht et al. 2019 remind us that ImageNet test examples tend to be simple, clear, close-up images, resulting in the current test set being perhaps too easy and not representative of harder images encountered in the real world. Geirhos et al. 2020a and Arjovsky et al. 2019 argue that image classification datasets contain "spurious cues" or "shortcuts." For instance, models may use an image's background to predict the foreground object's class; a cow tends to co-occur with a green pasture, and even though the background is inessential to the object's identity, models may predict "cow" primarily using the green pasture background cue. When datasets contain spurious cues, they can lead to performance estimates that are optimistic.

To counteract this, we curate two hard ImageNet test sets of adversarially filtered examples. By using adversarial filtration, we can test how well models perform when simple-to-classify examples are removed, including examples that are solved with simple spurious cues. Some adversarially filtered examples are depicted in figure 3.1, which are simple for humans but hard for models. Previously it has been shown that misclassified images can transfer between models; however, these demonstrations relied on synthetic distributions (Geirhos et al. 2018; Hendrycks

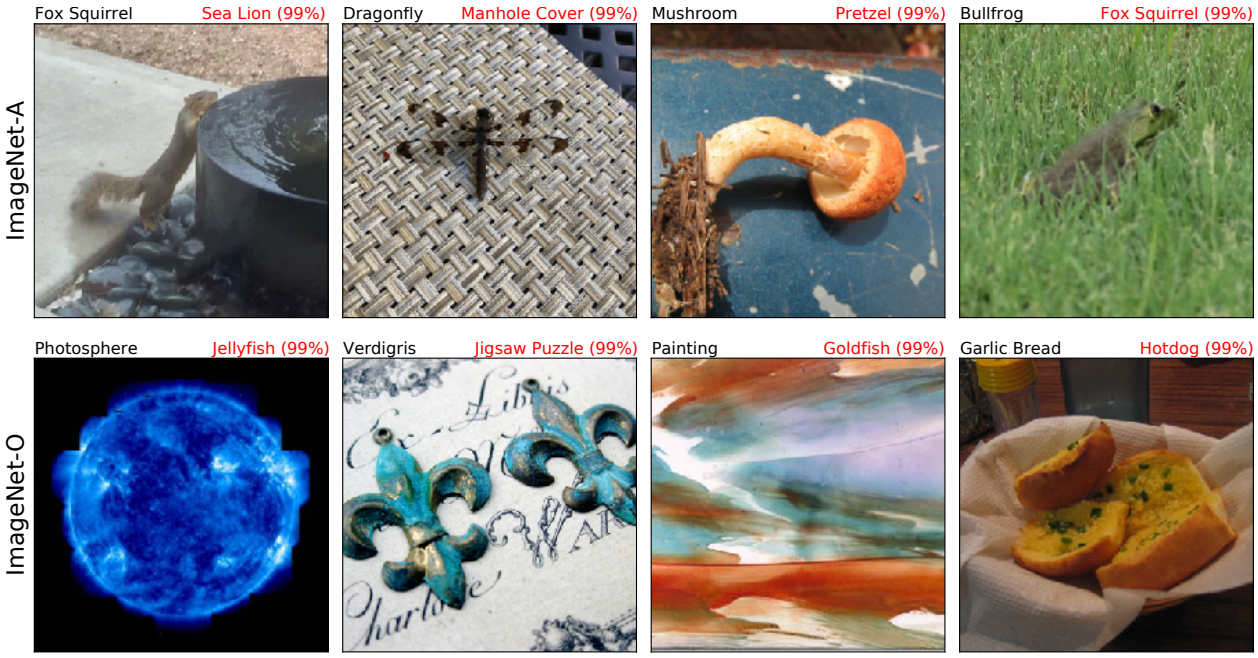


Figure 3.1: Examples from ImageNet-A and ImageNet-O. The black text is the actual class, and the red text is a ResNet-50 prediction and its confidence. ImageNet-A contains images that classifiers should be able to classify. ImageNet-O contains out-of-distribution anomalies of unforeseen classes which should result in low-confidence predictions. ImageNet-1K models do not train on examples from "Photosphere" nor "Verdigris" classes, so these images are out-of-distribution.

and Dietterich 2019) and adversarial distortions (Szegedy et al. 2013). Our examples demonstrate that is is possible to reliably fool many models with clean natural images highlighting a practical problem that needs to be addressed as opposed to a more theoretical or abstract problem.

We demonstrate that clean examples that we collected can reliably degrade and transfer to other unseen classifiers with our first dataset. Transfer in this setting means that the average performance of an unseen model is comparable (meaning not significantly better) to the model we used for filtering. This phenomenon of transferring is hard to classify. We call this dataset ImageNet-A, which contains images from a distribution unlike the ImageNet training distribution. ImageNet-A examples belong to ImageNet classes, but the examples are harder and transfer to other models. They cause consistent classification mistakes due to scene complications encountered in the long tail of scene configurations and by exploiting classifier blind spots (see Section 3.3.3). Since examples transfer reliably, this dataset shows models have previously unappreciated shared weaknesses.

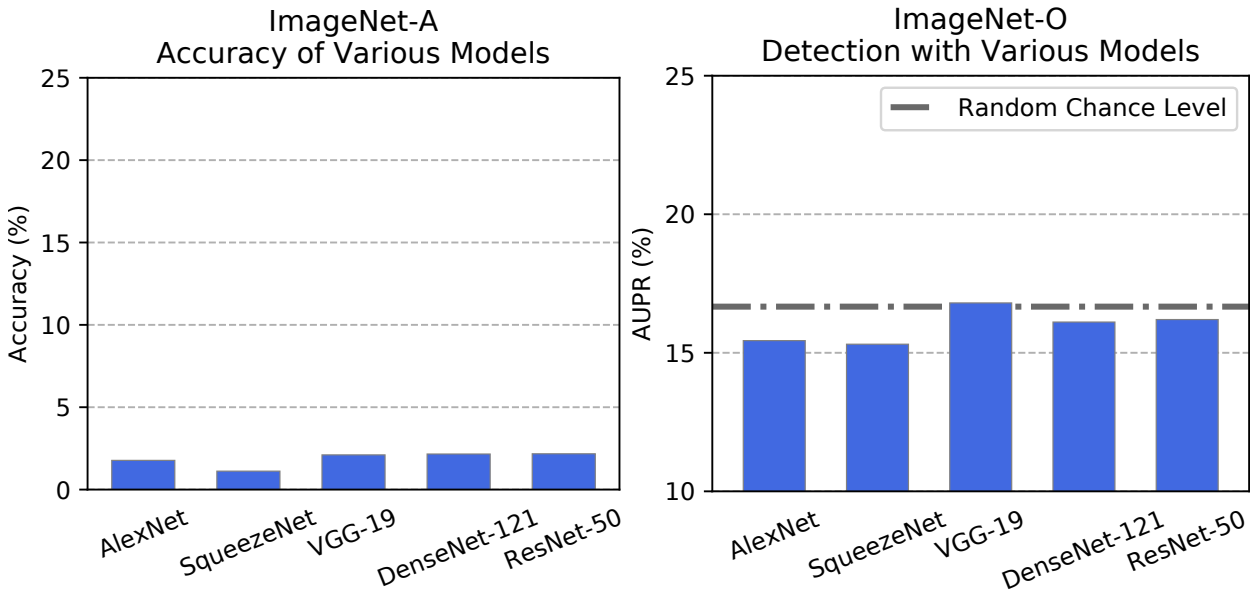


Figure 3.2: Various ImageNet classifiers of different architectures fail to generalize well to ImageNet-A and ImageNet-O. Higher Accuracy and higher AUPR is better. See 3.4 for a description of the AUPR out-of-distribution detection measure. These specific model parameters are unseen during adversarial filtration, so our adversarially filtered examples transfer across models.

The second dataset allows us to test model uncertainty estimates when semantic factors of the data distribution shift. Our second dataset is ImageNet-O, which contains image concepts from outside ImageNet-1K. These out-of-distribution images reliably cause models to mistake the examples as high-confidence in-distribution examples. To our knowledge this is the first dataset of anomalies or out-of-distribution examples developed to test ImageNet models. While ImageNet-A enables us to test image classification performance when the *input data distribution shifts*, ImageNet-O enables us to test out-of-distribution detection performance when the *label distribution shifts*.

We examine methods to improve performance on adversarially filtered examples. However, this is difficult because figure 3.2 shows that examples successfully transfer to unseen or black-box models. To improve robustness, numerous techniques have been proposed. We find data augmentation techniques such as adversarial training decrease performance, while others can help by a few percent. We also find that a $10\times$ increase in training data corresponds to a less than a 10% increase in accuracy. Finally, we show that improving model architectures is a promising avenue

toward increasing robustness. Even so, current models have substantial room for improvement.

3.2 Related Work

Adversarial Examples. Real-world images may be chosen adversarially to cause performance decline. I. Goodfellow et al. 2017 define adversarial examples (Szegedy et al. 2014) as "inputs to machine learning models that an attacker has intentionally designed to cause the model to make a mistake". Most adversarial examples research centers around artificial ℓ_p adversarial examples, which are examples perturbed by nearly worst-case distortions that are small in an ℓ_p sense. Attackers can reliably and easily create black-box attacks by exploiting these consistent *naturally occurring* model errors, and thus carefully applying gradient perturbations to create an artificial attack is unnecessary. This less restricted threat model has been discussed but not explored thoroughly before.

Several other forms of adversarial attacks have been considered in the literature, including elastic deformations (C. Xiao et al. 2018), adversarial coloring (Bhattad et al. 2019; Hosseini and Poovendran 2018), synthesis via generative models (Baluja and Fischer 2017; Song et al. 2018) and evolutionary search (Nguyen, Yosinski, and Clune 2015), among others. Other work has shown how to print 2D (Kurakin, I. J. Goodfellow, and Bengio 2017; Tom B Brown et al. 2017) or 3D (Sharif et al. 2016; Athalye et al. 2017) objects that fool classifiers. These existing adversarial attacks are all based on synthesized images or objects, and some have questioned whether they provide a reliable window into real-world robustness (Gilmer et al. 2018). Our examples are closer in spirit to the hypothetical adversarial photographer discussed in (Tom B. Brown et al. 2018), and by definition these adversarial photos occur in the real world.

Adversarial Examples. Some types of adversarial attacks eschew ℓ_p norm ball constraints completely. For instance, (Baluja and Fischer 2017) synthesize adversarial examples with generative adversarial networks, and (Song et al. 2018) use variational autoencoders as well. Unfortunately, these examples are often classifier-specific and do not transfer to new models. Meanwhile, ImageNet-A adversarially filtered examples transfer and successfully attack current architectures.

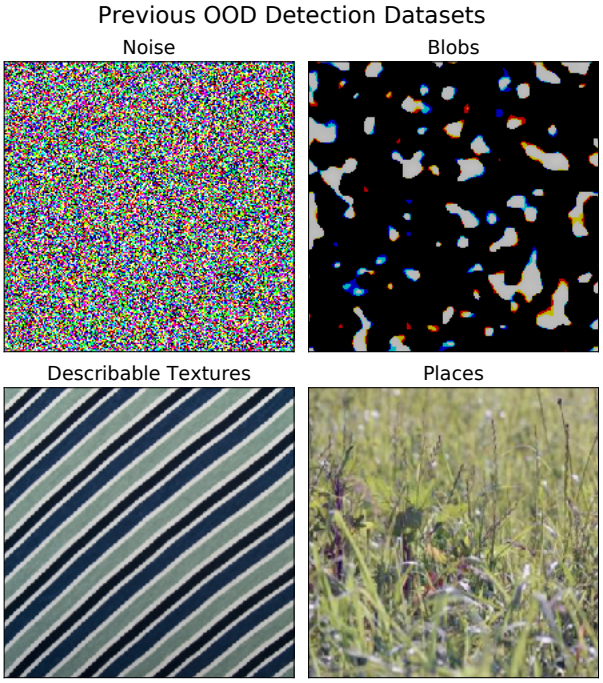


Figure 3.3: Previous work on out-of-distribution (OOD) detection uses synthetic anomalies and anomalies from wholly different data generating processes. For instance, previous work uses Bernoulli noise, blobs, the Describable Textures Dataset (Mircea Cimpoi et al. 2014), and Places365 scenes (B. Zhou et al. 2017) to test ImageNet out-of-distribution detectors. To our knowledge, we propose the first dataset of out-of-distribution examples collected for ImageNet models.

These adversarially filtered examples bear semblance to a theorized attack, the attack of the adversarial photographer (Tom B. Brown et al. 2018). This attacker is free to take a photograph of an image with choice over the camera viewpoint, so that the attack is free of the confines of the ℓ_p norm ball constraints. This attack has not been thoroughly studied empirically, but these could be thought of as a type of natural adversarial example. Bhattad et al. 2019; Hosseini and Poovendran 2018 attempt to color examples adversarially, and C. Xiao et al. 2018 create an adversarial elastic deformation. In both cases the modifications required to fool the network can become quite conspicuous (Kang et al. 2019) and artificial. Kurakin, I. J. Goodfellow, and Bengio 2017 show that ℓ_p adversarial examples can fool machine learning systems if they are carefully printed and if the perturbation is conspicuous. Meanwhile, adversarially filtered examples arise in nature and are not necessarily detectable by human vision alone. Nguyen, Yosinski, and Clune 2015 also violate norm ball constraints and create adversarial examples through evolutionary algorithms, but these adversarial examples are far out-of-distribution and are not naturally manifested in the real world.

Out-of-Distribution Detection. Generally, models learn a distribution, such as the ImageNet-1K distribution, and are tasked with producing quality anomaly scores that distinguish between usual test set examples and examples from held-out anomalous distributions. For instance, Hendrycks and Gimpel 2017 treat CIFAR-10 as the in-distribution and treat Gaussian noise and the SUN scene dataset (J. Xiao et al. 2010) as out-of-distribution data. That paper also shows that the negative of the maximum softmax probability, or the the negative of the classifier prediction probability, is a high-performing anomaly score that can separate in- and out-of-distribution examples, so much so that it remains competitive to this day. Since that time, other works on out-of-distribution detection continue to use datasets from other research benchmarks as stand-ins for out-of-distribution datasets. For example, some use the datasets shown in figure 3.3 as out-of-distribution datasets (Hendrycks, Mazeika, and Dietterich 2019). However, many of these anomaly sources are unnatural and deviate in numerous ways from the distribution of usual examples (Ahmed and Courville 2019). In fact, some of the distributions can be deemed anomalous from local image statistics alone. Meinke and Hein 2019 propose studying adversarial out-of-distribution detection by detecting adversarially optimized uniform noise. In contrast, we propose a dataset for more realistic adversarial anomaly detection; our dataset contains hard anomalies generated by shifting the distribution's labels and keeping non-semantic factors similar to the in-distribution.

Spurious Cues and Unintended Shortcuts. Models may learn spurious cues and obtain high accuracy but for the wrong reasons (Lapuschkin et al. 2019). Spurious cues are a studied problem in natural language processing (Cai, Tu, and Gimpel 2017; Gururangan et al. 2018). Many recently introduced datasets in NLP use adversarial filtration to create "adversarial datasets" by sieving examples solved with simple spurious cues (Sakaguchi et al. 2019; Bhagavatula et al. 2019; Zellers et al. 2019; Dua et al. 2019). Like this recent concurrent research, we also use adversarial filtration (Sung 1995), but the technique of adversarial filtration has not been applied to collecting image datasets before. Additionally, adversarial filtration in NLP uses filtration to remove only the easiest examples, while we use filtration to select only the hardest examples. Moreover, our examples transfer to weaker models, while in NLP the most used adversarial filtration technique AFLite (Sakaguchi

et al. 2019) does not produce examples that transfer to less performant models (Bisk et al. 2019). We show that adversarial filtration algorithms can find examples that automatically and reliably transfer to both simpler and stronger models. Since adversarial filtration can remove examples that are solved by simple spurious cues, models must learn more robust features for our datasets.

Robustness to Shifted Input Distributions. Recht et al. 2019 create a new ImageNet test set resembling the original test set as closely as possible. They found evidence that matching the difficulty of the original test set required selecting images deemed the easiest and most obvious by Mechanical Turkers. ImageNet-A helps measure generalization to harder scenarios. Brendel and Bethge 2018 show that classifiers that do not know the spatial ordering of image regions can be competitive on the ImageNet test set, possibly due to the dataset's lack of difficulty. Judging classifiers by their performance on easier examples has potentially masked many of their shortcomings. For example, Geirhos et al. 2019 artificially overwrite each ImageNet image's textures and conclude that classifiers learn to rely on textural cues and under-utilize information about object shape. Recent work shows that classifiers are highly susceptible to non-adversarial stochastic corruptions (Hendrycks and Dietterich 2019). While they distort images with 75 different algorithmically generated corruptions, our sources of distribution shift tend to be more heterogeneous and varied, and our examples are naturally occurring.

3.3 The Design and Construction of ImageNet-A and ImageNet-O

3.3.1 Design

ImageNet-A is a dataset of adversarially filtered examples for ImageNet classifiers, or real-world examples that fool current classifiers. To find adversarially filtered examples, we first download numerous images related to an ImageNet class. Thereafter we delete the images that fixed ResNet-50 (He et al. 2015) classifiers correctly predict. We chose ResNet-50 due to its widespread use. Later we show that examples which fool ResNet-50 transfer reliably to other unseen models. With the remaining incorrectly classified images, we manually select a subset of high-quality images.

Figure 3.4: Additional adversarially filtered examples from the ImageNet-A dataset. Examples are adversarially selected to cause classifier accuracy to degrade. The black text is the actual class, and the red text is a ResNet-50 prediction.

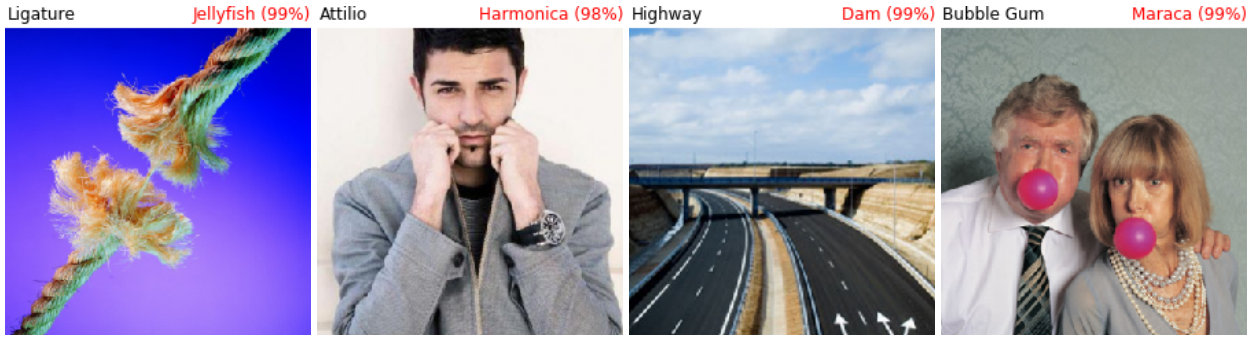


Figure 3.5: Additional adversarially filtered examples from the ImageNet-O dataset. Examples are adversarially selected to cause out-of-distribution detection performance to degrade. Examples do not belong to ImageNet classes, and they are wrongly assigned highly confident predictions. The black text is the actual class, and the red text is a ResNet-50 prediction and the prediction confidence.

Next, ImageNet-O is a dataset of adversarially filtered examples for ImageNet out-of-distribution detectors. To create this dataset, we download ImageNet-22K and delete examples from ImageNet-1K. With the remaining ImageNet-22K examples that do not belong to ImageNet-1K classes, we keep examples that are classified by a ResNet-50 as an ImageNet-1K class with high confidence. Then we manually select a subset of high-quality images.

Both datasets were manually labelled by graduate students over several months. This is because a large share of images in the ImageNet test set contain multiple classes per image (Stock and Cissé 2018). Therefore, producing a high-quality dataset without multilabel images can be challenging with usual annotation techniques. By high-quality we refer to both being a singleton and recognizable instance of the target class. To ensure images do not fall into more than one of

the several hundred classes, we had graduate students memorize the classes to build a high-quality test set.

ImageNet-A Class Restrictions. We select a 200-class subset of ImageNet-1K's 1,000 classes so that errors among these 200 classes would be considered egregious (Russakovsky et al. 2014). For instance, wrongly classifying Norwich terriers as Norfolk terriers does less to demonstrate faults in current classifiers than mistaking a Persian cat for a candle. We additionally avoid rare classes such as "snow leopard," classes that have changed much since 2012 such as "iPod," coarse classes such as "spiral," classes that are often image backdrops such as "valley," and finally classes that tend to overlap such as "honeycomb," "bee,'" "bee house," and "bee eater"; "eraser," "pencil sharpener" and "pencil case"; "sink," "medicine cabinet," "pill bottle" and "band-aid"; and so on. The 200 ImageNet-A classes cover most broad categories spanned by ImageNet-1K; see the Supplementary Materials A.1 for the full class list.

ImageNet-O Class Restrictions. We again select a 200-class subset of ImageNet-1K's 1,000 classes. These 200 classes determine the in-distribution or the distribution that is considered usual. The remaining 800 classes could be used as data for Outlier Exposure Hendrycks, Mazeika, and Dietterich 2019. As before, the 200 classes cover most broad categories spanned by ImageNet-1K; see the Supplementary Materials A.2 for the full class list.

3.3.2 Collection

ImageNet-A Data Aggregation. Curating a large set of adversarially filtered examples requires combing through an even larger set of images. Fortunately, the website iNaturalist has millions of user-labeled images of animals, and Flickr has even more user-tagged images of objects. We download images related to each of the 200 ImageNet classes by leveraging user-provided labels and tags. After exporting or scraping data from sites including iNaturalist, Flickr, and DuckDuckGo, we adversarially select images by removing examples that fail to fool our ResNet-50 models. Of the remaining images, we select low-confidence images and then ensure each image is valid through human review. For this procedure to work, many images are necessary; if we only

used the original ImageNet test set as a source rather than iNaturalist, Flickr, and DuckDuckGo, some classes would have zero images after the first round of filtration.

For concreteness, we describe the selection process for the dragonfly class. We download 81,413 dragonfly images from iNaturalist, and after performing a basic filter with ResNet-50, we have 8,925 dragonfly images. In the algorithmically suggested shortlist, 1,452 images remain. From this shortlist, 80 dragonfly images are manually selected, but hundreds more could be chosen. Hence for just one class we may review over 1,000 images.

We now describe this process in more detail. We use a small ensemble of ResNet-50s for filtering, one pre-trained on ImageNet-1K then fine-tuned on the 200 class subset, and one pre-trained on ImageNet-1K where 200 of its 1,000 logits are used in classification. Both classifiers have similar accuracy on the 200 clean test set classes from ImageNet-1K. The ResNet-50s perform 10-crop classification of each image, and should any crop be classified correctly by the ResNet-50s, the image is removed. If either ResNet-50 assigns greater than 15% confidence to the correct class, the image is also removed; this is done so that adversarially filtered examples yield misclassifications with low confidence in the correct class, like in untargeted adversarial attacks. Now, some classification confusions are greatly over-represented, such as Persian cat and lynx. We would like ImageNet-A to have great variability in its types of errors and cause classifiers to have a dense confusion matrix. Consequently, we perform a second round of filtering to create a shortlist where each confusion only appears at most 15 times. Finally, we manually select images from this shortlist in order to ensure ImageNet-A images are simultaneously valid, single-class, and high-quality. In all, the ImageNet-A dataset has 7,500 adversarially filtered examples.

ImageNet-O Data Aggregation. Our dataset for adversarial out-of-distribution detection is created by fooling ResNet-50 out-of-distribution detectors. The negative of the prediction confidence of a ResNet-50 ImageNet classifier serves as our anomaly score (Hendrycks and Gimpel 2017). Usually in-distribution examples produce higher confidence predictions than OOD examples, but we curate OOD examples that have high confidence predictions. To gather candidate adversarially filtered examples, we use the ImageNet-22K dataset with ImageNet-1K

classes deleted. We choose the ImageNet-22K dataset since it was collected in the same way as ImageNet-1K. ImageNet-22K allows us to have coverage of numerous visual concepts and vary the distribution's semantics without unnatural or unwanted non-semantic data shift. After excluding ImageNet-1K images, we process the remaining ImageNet-22K images and keep the images which cause the ResNet-50 to have high confidence, or a low anomaly score. We then manually select a high-quality subset of the remaining images to create ImageNet-O. We suggest only training models with data from the 1,000 ImageNet-1K classes, since the dataset becomes trivial if models train on ImageNet-22K. To our knowledge, this dataset is the first anomalous dataset curated for ImageNet models and enables researchers to study adversarial out-of-distribution detection. The ImageNet-O dataset has 2,000 adversarially filtered examples since anomalies are rarer; this has the same number of examples per class as ImageNetV2 (Recht et al. 2019). Thus we use adversarial filtration to select examples that are difficult for a fixed ResNet-50, and we will show these examples straightforwardly transfer to unseen models. Additional example ImageNet-O images are in 3.5.

For the collection of ImageNet-O we refrain from testing with classification models with a background class. While there are many classification tasks which incorporate a "background" class into the final predictions such as many segmentation datasets (Lin et al. 2014; Everingham et al. 2009), most of the common multi-class classification datasets do not include such a category. Most importantly the ImageNet dataset does not include a background class which is what our dataset aims to test models from. There still remains the potential issue that if a model had been trained with a background class then the images from ImageNet-O would be classified as such. It has been shown that depending on the score utilized such as from generative models out-of-distribution examples can actually achieve greater in-distribution (or lower anomaly) score in Hendrycks and Gimpel 2017. We also show in a different chapter how poorly background classes perform in distinguishing out from in-distribution see 4.3.

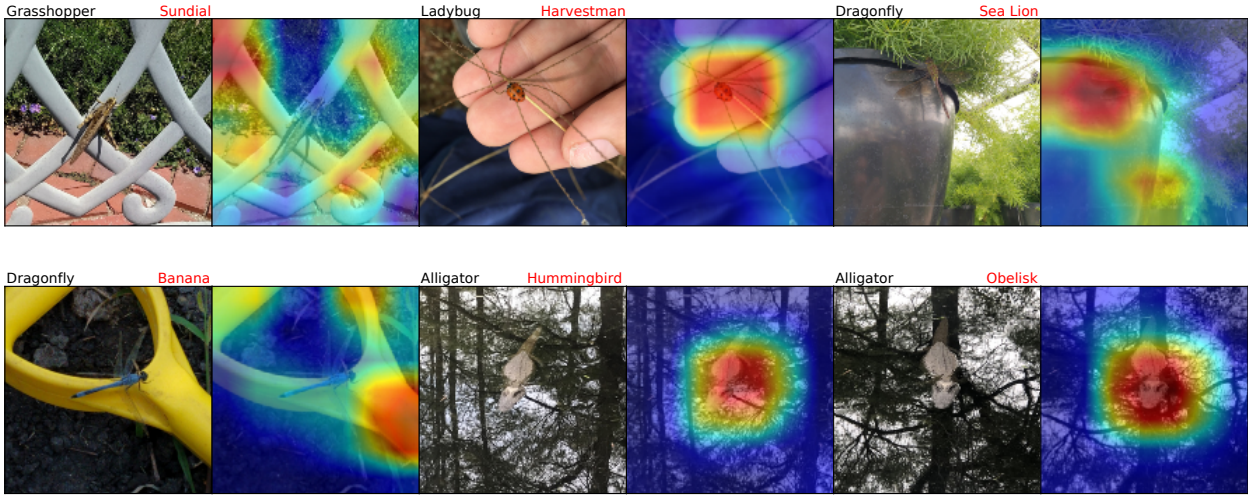


Figure 3.6: Examples from ImageNet-A demonstrating classifier failure modes. Adjacent to each natural image is its heatmap generated by "gradCAM" (Selvaraju, Das, et al. 2019). The heatmap is generated by taking the classification prediction and computing the gradients from the given output prediction back to the image. The gradients are scaled to -1 to 1 range before adding back to the original image. The -1 implies negative influence to the prediction and is depicted as blue, while 1 implies positive influence for the prediction and is the depicted as red. The resulting image is rescaled after adding the image gradient. Classifiers may use erroneous background cues for prediction and the technique gradCAM can be used. Further description of these failure modes is in Section 3.3.3.

3.3.3 Illustrative Failure Modes

Examples in ImageNet-A uncover numerous failure modes of modern convolutional neural networks. We describe our findings after having viewed tens of thousands of candidate adversarially filtered examples. Some of these failure modes may also explain poor ImageNet-O performance, but for simplicity we describe our observations with ImageNet-A examples.

Observe figure 3.6. The first two images suggest models may overgeneralize visual concepts. It may confuse metal with sundials, or thin radiating lines with harvestman bugs. We also observed that networks overgeneralize tricycles to bicycles and circles, digital clocks to keyboards and calculators, and more. We also observe that models may rely too heavily on color and texture, as shown with the dragonfly images. Since classifiers are taught to associate entire images with an object class, frequently appearing background elements may also become associated with a class, such as wood being associated with nails. Other examples include classifiers heavily associating hummingbird feeders with hummingbirds, leaf-covered tree branches being associated with the

white-headed capuchin monkey class, snow being associated with shovels, and dumpsters with garbage trucks. Additionally figure 3.6 shows an American alligator swimming. With different frames, the classifier prediction varies erratically between classes that are semantically loose and separate. For other images of the swimming alligator, classifiers predict that the alligator is a cliff, lynx, and a fox squirrel. Current convolutional networks have pervasive and diverse failure modes that are tested with ImageNet-A.

3.4 Experiments

We show that adversarially filtered examples collected to fool fixed ResNet-50 models reliably transfer to other models, indicating that current convolutional neural networks have shared weaknesses and failure modes. In the following sections, we analyze whether robustness can be improved by using data augmentation, using more real labeled data, and using different architectures. For the first two sections, we analyze performance with a fixed architecture for comparability, and in the final section we observe performance with different architectures. As a preliminary, we define our metrics.

Metrics. Our metric for assessing robustness to adversarially filtered examples for classifiers is the top-1 *accuracy* on ImageNet-A. For reference, the top-1 accuracy on the 200 ImageNet-A classes using usual ImageNet images is usually greater than or equal to 90% for ordinary classifiers.

Our metric for assessing out-of-distribution detection performance of ImageNet-O examples is the area under the precision-recall curve (*AUPR*). This metric requires anomaly scores. Our anomaly score is the negative of the maximum softmax probabilities (Hendrycks and Gimpel 2017) from a model that can classify the 200 ImageNet-O classes specified in Section 3.3. We collect anomaly scores with the ImageNet validation examples for the said 200 classes. Then, we collect anomaly scores for the ImageNet-O examples. Higher performing OOD detectors would assign ImageNet-O examples lower confidences, or higher anomaly scores. With these anomaly scores, we can compute the area under the precision-recall curve (Saito and Rehmsmeier 2015). Random chance levels for the AUPR is approximately 16.67% with ImageNet-O, and the maximum AUPR

is 100%.

Data Augmentation. We examine popular data augmentation techniques and note their effect on robustness. In this section we exclude ImageNet-O results, as the data augmentation techniques hardly help with out-of-distribution detection as well. As a baseline, we train a new ResNet-50 from scratch and obtain 2.17% accuracy on ImageNet-A. Now, one purported way to increase robustness is through adversarial training, which makes models less sensitive to ℓ_p perturbations. We use the adversarially trained model from (Wong, Rice, and Kolter 2020), but accuracy decreases to 1.68%. Next, Geirhos et al. 2019 propose making networks rely less on texture by training classifiers on images where textures are transferred from art pieces. They accomplish this by applying style transfer to ImageNet training images to create a stylized dataset, and models train on these images. While this technique is able to greatly increase robustness on synthetic corruptions (Hendrycks and Dietterich 2019), Style Transfer increases ImageNet-A accuracy by 0.13% over the ResNet-50 baseline. A recent data augmentation technique, AugMix (Hendrycks, Mu, et al. 2020), which takes linear combinations of different data augmentations increases accuracy to 3.8%. Cutout augmentation (Devries and Taylor 2017) randomly occludes image regions and corresponds to 4.4% accuracy. Moment Exchange (MoEx) (Li et al. 2020) exchanges feature map moments between images, and this increases accuracy to 5.5%. Mixup (Hongyi Zhang et al. 2017) trains networks on elementwise convex combinations of images and their interpolated labels; this technique increases accuracy to 6.6%. CutMix (Yun et al. 2019) superimposes images regions within other images and yields 7.3% accuracy. At best these data augmentations techniques improve accuracy by approximately 5% over the baseline. Results are summarized in figure 3.7.

More Labeled Data. One possible explanation for consistently low ImageNet-A accuracy is that all models are trained only with ImageNet-1K, and using additional data may resolve the problem. To test this hypothesis we pre-train a ResNet-50 on Places365 (B. Zhou et al. 2017), a large-scale scene recognition dataset. After fine-tuning the Places365 model on ImageNet-1K, we find that accuracy is 1.56%. Consequently, even though scene recognition models are purported to

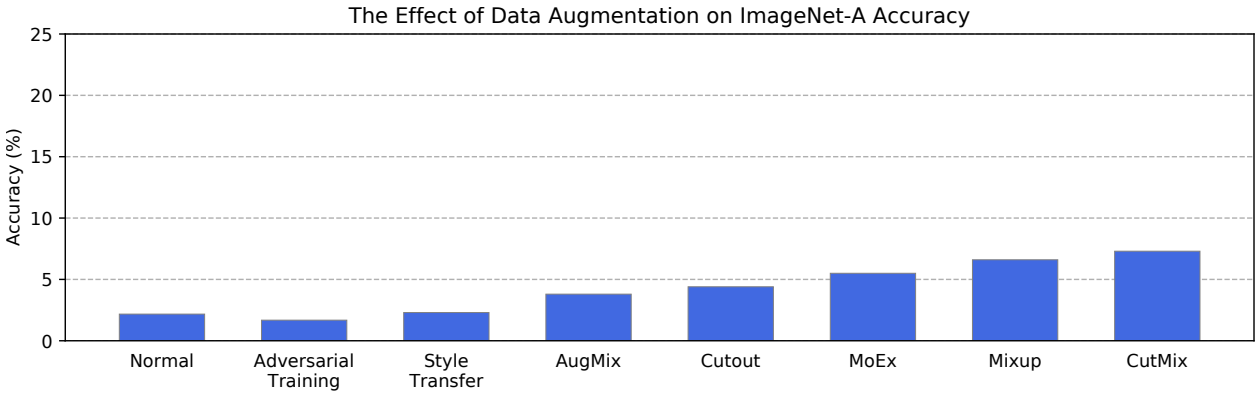


Figure 3.7: Some data augmentation methods can slightly improve performance.

have qualitatively distinct features (B. Zhou et al. 2019), this is not enough to improve ImageNet-A performance. Likewise, Places365 pre-training does not improve ImageNet-O detection, as its AUPR is 14.88%. Next, we see whether labeled data from ImageNet-A itself can help. We take baseline ResNet-50 with 2.17% ImageNet-A accuracy and fine-tune it on 80% of ImageNet-A. This leads to no clear improvement on the remaining 20% of ImageNet-A since the top1 and top5 accuracies are below 2% and 5% respectively. Last, we pre-train using an order of magnitude more training data with ImageNet-21K. This dataset contains approximately 21,000 classes and approximately 14 million images. To our knowledge this is the largest publicly available database of labeled natural images. Using a ResNet-50 pretrained on ImageNet-21K, we fine-tune the model on ImageNet-1K and attain 11.41%accuracy on ImageNet-A, a 9.24% increase. Likewise, the AUPR for ImageNet-O improves from 16.20% to 21.86%, although this improvement is less significant since ImageNet-O images overlap with ImageNet-21K images. Overall, an order of magnitude increase in labeled training data can provide some improvements in accuracy.

Architectural Changes. We find that model architecture can play a large role in ImageNet-A accuracy and ImageNet-O detection performance. Simply increasing the width and number of layers of a network is sufficient to automatically impart more ImageNet-A accuracy and ImageNet-O OOD detection performance. Increasing network capacity has been shown to improve performance on ℓ_p adversarial examples (Kurakin, I. Goodfellow, and Bengio 2017), common corruptions (Hendrycks and Dietterich 2019), and now also improves performance for adversarially filtered images. For example, a ResNet-50's top-1 accuracy and AUPR is 2.17% and

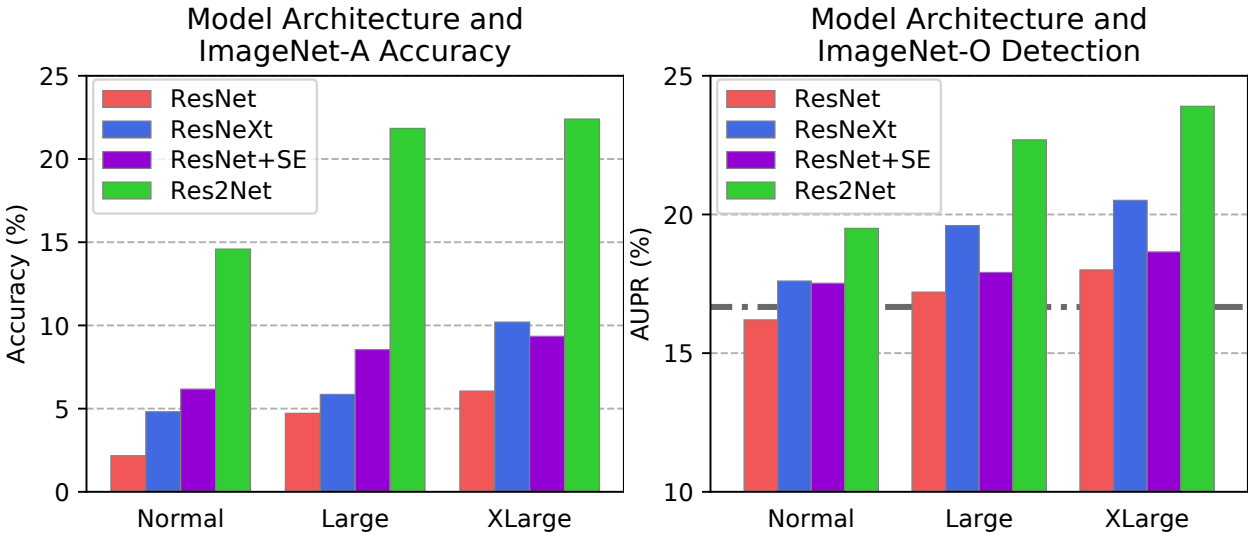


Figure 3.8: Increasing model size and other architecture changes can greatly improve performance. Note Res2Net and ResNet+SE have a ResNet backbone. Normal model sizes are ResNet-50 and ResNeXt-50 (32×4d), Large model size are ResNet-101 and ResNeXt-101 (32×4d), and XLarge Model sizes are ResNet-152 and (32×8d).

16.2%, respectively, while a ResNet-152 obtains 6.1% top-1 accuracy and 18.0% AUPR. Another architecture change that reliably helps is using the grouped convolutions found in ResNeXts (Xie et al. 2016). A ResNeXt-50 (32×4d) obtains a 4.81% top1 ImageNet-A accuracy and a 17.60% ImageNet-O AUPR. Another architectural change is self attention. Convolutional neural networks with self-attention (Hu et al. 2018) are designed to better capture long-range dependencies and interactions across an image. We consider the self-attention technique called Squeeze-and-Excitation (SE) (Hu, Shen, and Sun 2018), which won the final ImageNet competition in 2017. A ResNet-50 with Squeeze-and-Excitation attains 6.17% accuracy. However, for larger ResNets, self-attention does little to improve ImageNet-O detection. Finally, we consider the ResNet-50 architecture with its residual blocks exchanged with recently introduced Res2Net v1b blocks (Gao et al. 2019). This change increases accuracy to 14.59% and the AUPR to 19.5%. A ResNet-152 with Res2Net v1b blocks attains 22.4% accuracy and 23.9% AUPR. Compared to data augmentation or an order of magnitude more labeled training data, some architectural changes can provide far more robustness gains. Consequently future improvements to model architectures is a promising path towards greater robustness.

3.5 Conclusion

In this chapter, we introduced adversarially filtered examples for image classifiers and out-of-distribution detectors. Our ImageNet-A dataset degrades classification accuracy across known classifiers, and it measures robustness to input data distribution shifts. Likewise, ImageNet-O adversarially filtered examples reliably degrade ImageNet out-of-distribution detection performance, and it measures robustness to label distribution shifts. ImageNet-O enables the measurement of adversarial out-of-distribution detection performance, and is the first ImageNet out-of-distribution detection dataset. Our adversarial filtration process removes examples solved by simple spurious cues, so our datasets enable researchers to observe performance when simple spurious cues are removed. Our naturally occurring images expose common blindspots of current convolutional networks, and solving these tasks will require addressing long-standing but under-explored failure modes of current models such as over-reliance on texture, over-generalization, and spurious cues. We found that these failures are slightly less pronounced with different data augmentation strategies. However, we identified that architectural improvements can provide large gains in model robustness, and there is much room for future research. In this work, we introduce two new and difficult ImageNet test sets to measure model performance under distribution shift—an important research aim as computer vision systems are deployed in increasingly precarious real-world environments.

CHAPTER 4

SCALING ANOMALY DETECTION TO LARGE SCALE IMAGES

4.1 Overview

Detecting out-of-distribution inputs is important in real-world applications of deep learning. When faced with anomalous inputs flagged as such, systems may initiate a conservative fallback policy or defer to human judgment. This is especially important in safety-critical applications of deep learning, such as self-driving cars or medical applications. Accordingly, research on out-of-distribution detection has a rich history spanning several decades (Schölkopf et al. 1999b; M. M. Breunig et al. 2000; A. Emmott et al. 2015). Recent work leverages deep neural representations for out-of-distribution detection in complex domains, such as image data (Hendrycks and Gimpel 2017; Lee et al. 2018a; Hendrycks, Mazeika, and Dietterich 2019). However, these works still primarily use small-scale datasets with low-resolution images and few classes. As the community moves towards more realistic, large-scale settings, strong baselines and high-quality benchmarks are imperative for future progress.

In addition to focusing on small-scale datasets, previous formulations of anomaly detection treat entire images as anomalies. In practice, an image could be anomalous in localized regions while being in-distribution elsewhere. Knowing which regions of an image are anomalous could allow for safer handling of unfamiliar objects in the case of self-driving cars. Creating a benchmark for this task is difficult, though, as simply cutting and pasting anomalous objects into images introduces various unnatural giveaway cues such as edge effects, mismatched orientation, and lighting, all of which trivialize the task of anomaly segmentation (Blum et al. 2019).

To overcome these issues, we utilize a simulated driving environment to create the novel StreetHazards dataset for anomaly segmentation. Using the Unreal Engine and the open-source CARLA simulation environment (Dosovitskiy et al. 2017), we insert a diverse array of foreign objects into driving scenes and re-render the scenes with these novel objects. This enables integration of the foreign objects into their surrounding context with correct lighting and orientation.

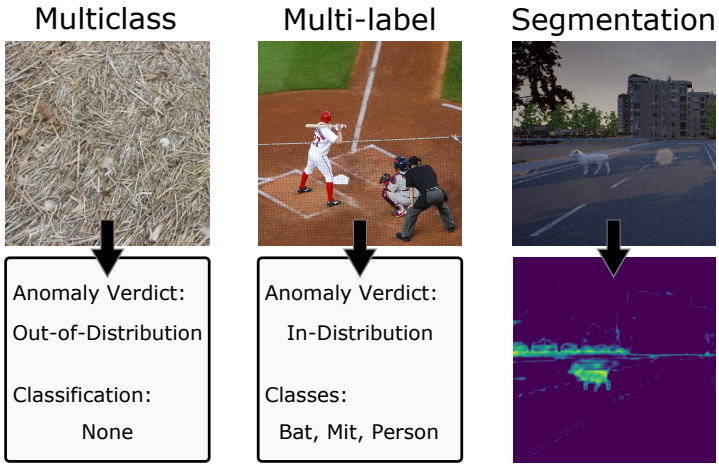


Figure 4.1: We scale up out-of-distribution detection to large-scale multi-class datasets with hundreds of classes, multi-label datasets with complex scenes, and anomaly segmentation in driving environments. In all three settings, we find that an OOD detector based on the maximum logit outperforms previous methods, establishing a strong and versatile baseline for future work on large-scale OOD detection.

To complement the StreetHazards dataset, we convert the BDD100K semantic segmentation dataset (Yu et al. 2018) into an anomaly segmentation dataset, which we call BDD-Anomaly. By leveraging the large scale of BDD100K, we reserve infrequent object classes to be anomalies. We combine this dataset with StreetHazards to form the Combined Anomalous Object Segmentation (CAOS) benchmark. The CAOS benchmark improves over previous evaluations for anomaly segmentation in driving scenes by evaluating detectors on realistic and diverse anomalies. We evaluate several baselines on the CAOS benchmark and discuss problems with porting existing approaches from earlier formulations of out-of-distribution detection.

In more traditional whole-image anomaly detection, large-scale datasets such as ImageNet (Deng et al. 2009) and Places365 (B. Zhou et al. 2017) present unique challenges not seen in small-scale settings, such as a plethora of fine-grained object classes. We demonstrate that the MSP detector, a state-of-the-art method for small-scale problems, does not scale well to these challenging conditions. Moreover, in the common real-world case of multi-label data, the MSP detector cannot naturally be applied in the first place, as it requires softmax probabilities.

Through extensive experiments, we identify a detector based on the maximum logit (MaxLogit) that greatly outperforms strong baselines in large-scale multi-class, and anomaly segmentation set-

tings. In each of these three settings, we discuss why MaxLogit provides superior performance, and we show that these gains are hidden if one looks at small-scale problems alone. The code for our experiments and the CAOS benchmark datasets are available at github.com/hendrycks/anomaly-seg.

4.2 Related Work

Anomaly Segmentation. Several prior works explore segmenting anomalous image regions. One line of work uses the WildDash dataset (Zendel et al. 2018), which contains numerous annotated driving scenes in conditions such as snow, fog, and rain. The WildDash test set contains fifteen "negative image" from different domains for which the goal is to mark the entire image as out-of-distribution. Thus, while the task is segmentation, the anomalies do not exist as objects within an otherwise in-distribution scene. This setting is similar to that explored by (Hendrycks and Gimpel 2017), in which whole images from other datasets serve as out-of-distribution examples.

To approach anomaly segmentation on WildDash, (Krešo et al. 2018) train on multiple semantic segmentation domains and treat regions of images from the WildDash driving dataset as out-of-distribution if they are segmented as regions from different domains, i.e. indoor classes. (Bevandić et al. 2018) use ILSVRC 2012 images and train their network to segment the entirety of these images as out-of-distribution.

In medical anomaly segmentation and product fault detection, anomalies are regions of otherwise in-distribution images. (Baur et al. 2019) segment anomalous regions in brain MRIs using pixel-wise reconstruction loss. Similarly, (Haselmann, Gruber, and Tabatabai 2018) perform product fault detection using pixel-wise reconstruction loss and introduce an expansive dataset for segmentation of product faults. In these relatively simple domains, reconstruction-based approaches work well. In contrast to medical anomaly segmentation and fault detection, we consider complex images from street scenes. These images have high variability in scene layout and lighting, and hence are less amenable to reconstruction-based techniques.

The two works closest to our own are the Lost and Found (Pinggera et al. 2016) and Fishyscapes (Blum et al. 2019) datasets. In table 4.1, we quantitatively compare the CAOS benchmark to these

	Fishyscapes	Lost and Found	BDD-Anomaly (Ours)	StreetHazards (Ours)
Train Images	0	1036	6280	5125
Test Images	1000	1068	810	1500
Anomaly Types	12	9	3	250

Table 4.1: Quantitative comparison of the CAOS benchmark with related datasets. The BDD-Anomaly dataset treats three categories as anomalous and has many unique object instances within those categories. By contrast, Lost and Found has the same objects in multiple images and has only nine unseen objects at test time. StreetHazards leverages a simulated environment to naturally insert hundreds of varied anomalies.

datasets. The Lost and Found dataset consists of real images in a driving environment with small road hazards. The images were collected to mirror the Cityscapes dataset (Cordts et al. 2016) but are only collected from one city and so have less diversity. The dataset contains 35 unique anomalous objects, and methods are allowed to train on many of these. For Lost and Found, only nine unique objects are truly unseen at test time. Crucially, this is a different evaluation setting from our own, where anomalous objects are not revealed at training time, so their dataset is not directly comparable. Nevertheless, the BDD-Anomaly dataset fills several gaps in Lost and Found. First, the images are more diverse, because they are sourced from a more recent and comprehensive semantic segmentation dataset. Second, the anomalies are not restricted to small, sparse road hazards. Concretely, anomalous regions in Lost and Found take up 0.11% of the image on average, whereas anomalous regions in the BDD-Anomaly dataset are larger and fill 0.83% of the image on average. Finally, although the BDD-Anomaly dataset treats three categories as anomalous, compared to Lost and Found it has far more unique anomalous objects.

The Fishyscapes benchmark for anomaly segmentation consists of cut-and-paste anomalies from out-of-distribution domains. This is problematic, because the anomalies stand out as clearly unnatural in context. For instance, the orientation of anomalous objects is unnatural, and the lighting of the cut-and-paste patch differs from the lighting in the original image, providing an unnatural cue to anomaly detectors that would not exist for real anomalies. Techniques for detecting image manipulation (P. Zhou et al. 2018) are competent at detecting artificial image elements of this kind. Our StreetHazards dataset overcomes these issues by leveraging a simulated

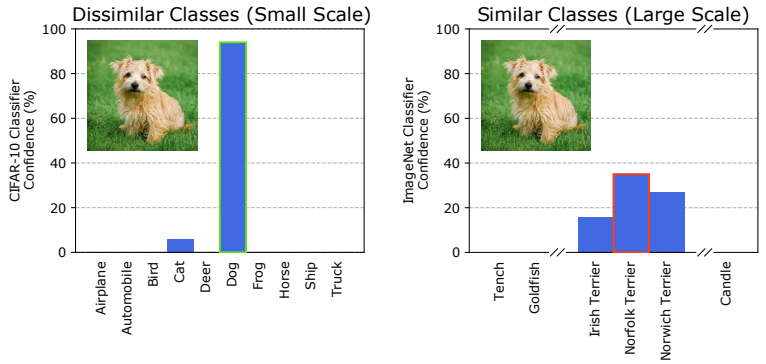


Figure 4.2: Small-scale datasets such as CIFAR-10 have relatively disjoint classes, but larger-scale datasets including ImageNet have several classes with high visual similarity to other classes. The implication is that large-scale classifiers disperse probability mass among several classes. If the prediction confidence is used for out-of-distribution detection, then images which have similarities to other classes will often wrongly be deemed out-of-distribution due to dispersed confidence. The dog is lower resolution for the CIFAR-10 classifier.

driving environment to naturally insert anomalous *3D models* into a scene rather than overlaying 2D images. These anomalies are integrated into the scene with proper lighting and orientation, mimicking real-world anomalies and making them significantly more difficult to detect.

Multi-Class Out-of-Distribution Detection. A recent line of work leverages deep neural representations from multi-class classifiers to perform out-of-distribution (OOD) detection on high-dimensional data, including images, text, and speech data. Hendrycks and Gimpel 2017 formulate the task and propose the simple baseline of using the maximum softmax probability of the classifier on an input to gauge whether the input is out-of-distribution. In particular, they formulate the task as distinguishing between examples from an in-distribution dataset and various out-of-distribution datasets. Importantly, entire images are treated as out-of-distribution.

Continuing this line of work, Lee et al. 2018a propose to improve the neural representation of the classifier to better separate in- and out-of-distribution examples. They use generative adversarial networks to produce near-distribution examples. In training, they encourage the classifier to output a uniform posterior on these synthetic out-of-distribution examples. Hendrycks, Mazeika, and Dietterich 2019 observe that outliers are often easy to obtain in large quantity from diverse, realistic datasets and demonstrate that training out-of-distribution detectors to detect these outliers generalizes to completely new, unseen classes of anomalies. Other work investigates

improving the anomaly detectors themselves given a fixed classifier (DeVries and Taylor 2018; Liang, Li, and Srikant 2018). However, as observed in Hendrycks, Mazeika, and Dietterich 2019, most of these works tune hyperparameters on a particular type of anomaly that is also seen at test time, so their evaluation setting is more lenient. We ensure that all anomalies seen at test time come from entirely unseen categories and are not tuned on in any way, hence we do not compare to techniques such as Liang, Li, and Srikant 2018. Additionally, in a point of departure from prior work, we focus primarily on large-scale images and datasets with many classes.

4.3 Multi-Class Prediction for OOD Detection

Problem with existing baselines. In large-scale image classification, a network is often tasked with predicting an object's identity from one of hundreds or thousands of classes, where class distinctions tend to be more fine and subtle. An increase in similarity and overlap between classes spells a problem for the multi-class out-of-distribution baseline (Hendrycks and Gimpel 2017). This baseline uses the negative maximum softmax probability as the anomaly score, or $-\max_k p(y = k \mid x)$. Classifiers tend to have higher confidence on in-distribution examples than out-of-distribution examples, enabling OOD detection. Assuming single-model evaluation and no access to other anomalies or test-time adaptation, the maximum softmax probability (MSP) is the state-of-the-art multi-class out-of-distribution detection method. However, we show that the MSP is problematic for large-scale datasets with many classes including ImageNet-1K and Places365 (B. Zhou et al. 2017). Probability mass can be dispersed among visually similar classes, as shown in figure 4.2. Consequently, a classifier may produce a low confidence prediction for an in-distribution image, not because the image is unfamiliar or out-of-distribution, but because the object's exact class is difficult to determine. To circumvent this problem, we propose using the negative of the maximum unnormalized logit for an anomaly score, which we call MaxLogit.

The MaxLogit has several benefits over the previous baseline of MSP. Empirically we find that MaxLogit outperforms the MSP, although the difference is marginal in small scale image datasets such as CIFAR-10 but show a larger improvement on CAOS benchmark and ImageNet see 4.2. The

two main reasons other than emperical performance to switch from MSP to MaxLogit are that the MaxLogit can work even if it is not a distributions and MaxLogit is unaffected by the number of classes. To expand upon each point, for several image tasks the output comes in the form of multi-label categories which the results are not from a softmax and not a distribution as the results do not sum to one, and MaxLogit does not require such a condition to in order to operate. The other point is a bit more subtle but none-the-less important. While the softmax operation does not change the maximum score, and does not change the ordering of classes per item, it does affect the relative ordering across items. This is because the softmax operation will convert the absolute differences per item in relative differences which loses information when comparing across items.

Datasets. To evaluate the MSP baseline out-of-distribution detector and the MaxLogit detector, we use ImageNet-1K object recognition dataset and Places365 scene recognition dataset as in-distribution datasets $\mathcal{D}_{\text{in}}$. We use several out-of-distribution test datasets $\mathcal{D}_{\text{out}}$, all of which are unseen during training. The first out-of-distribution dataset is *Gaussian* noise, where each pixel of these out-of-distribution examples are i.i.d. sampled from $\mathcal{N}(0, 0.5)$ and clipped to be contained within $[-1, 1]$. Another type of test-time noise is *Rademacher* noise, in which each pixel is i.i.d. sampled from $2 \cdot \text{Bernoulli}(0.5) - 1$, i.e. each pixel is 1 or -1 with equal probability. *Blob* examples are more structured than noise; they are algorithmically generated blob images. Meanwhile, *Textures* is a dataset consisting in images of describable textures (M. Cimpoi et al. 2014). When evaluating the ImageNet-1K detector, we use *LSUN* images, which is a dataset for scene recognition (Yu et al. 2015). Our final $\mathcal{D}_{\text{out}}$ is *Places69*, a scene classification dataset that does not share classes with Places365. In all, we evaluate against out-of-distribution examples spanning synthetic and realistic images.

Results. Results are shown in table 4.2. Observe that the proposed MaxLogit method outperforms the maximum softmax probability baseline for all three metrics on both ImageNet and Places365. These results were computed using a ResNet-50 trained on either ImageNet-1K or Places365. In the case of Places365, the AUROC improvement is over 10%. We note that the utility of the maximum logit could not as easily be appreciated in previous work's small-scale settings.

	FPR95 ↓			AUROC ↑			AUPR ↑		
$\mathcal{D}_{\text{in}}$	MSP	MaxLogit	KL	MSP	MaxLogit	KL	MSP	MaxLogit	KL
ImageNet	42.42	35.77	36.22	84.60	87.20	87.29	48.26	45.68	37.32
Places365	52.68	36.6	49.14	75.67	85.9	80.01	8.13	19.2	24.61

Table 4.2: Multi-class out-of-distribution detection results using the maximum softmax probability, maximum logit basline and KL Divergence between predicted and posterior. Results are on ImageNet and Places365. Values are rounded so that 99.995% rounds to 100%. Full results on individual $\mathcal{D}_{\text{out}}$ datasets and additional baselines are in the supplementary material.

For example, using the small-scale CIFAR-10 setup of (Hendrycks, Mazeika, and Dietterich 2019), the MSP attains an average AUROC of 90.08% while the maximum logit attains 90.22%, a 0.14% difference. However, in a large-scale setting, the difference can be over 10% on individual $\mathcal{D}_{\text{out}}$ datasets. We are not claiming that utilizing the maximum logit is a mathematically innovative formulation, only that it serves as a consistently powerful baseline for large-scale settings that went unappreciated in small-scale settings. In consequence, we suggest using the maximum logit as a new baseline for large-scale multi-class out-of-distribution detection.

4.4 The CAOS Benchmark

The Combined Anomalous Object Segmentation (CAOS) benchmark is comprised of two complementary datasets for evaluating anomaly segmentation systems on diverse, realistic anomalies. First is the StreetHazards dataset, which leverages simulation to provide a large variety of anomalous objects realistically inserted into driving scenes. Second is the BDD-Anomaly dataset, which consists of real images taken from the BDD100K dataset (Yu et al. 2018). StreetHazards contains a highly diverse array of anomalies; BDD-Anomaly contains anomalies in real-world images. Together, these datasets allow researchers to judge techniques on their ability to segment diverse anomalies as well as anomalies in real images. All images have 720×1280 resolution, and we recommend evaluating with the AUROC, AUPR, and FPRK metrics, which we describe in Section 4.4.1.

The StreetHazards Dataset. StreetHazards is an anomaly segmentation dataset that leverages simulation to provide diverse, realistically-inserted anomalous objects. To create the StreetHazards

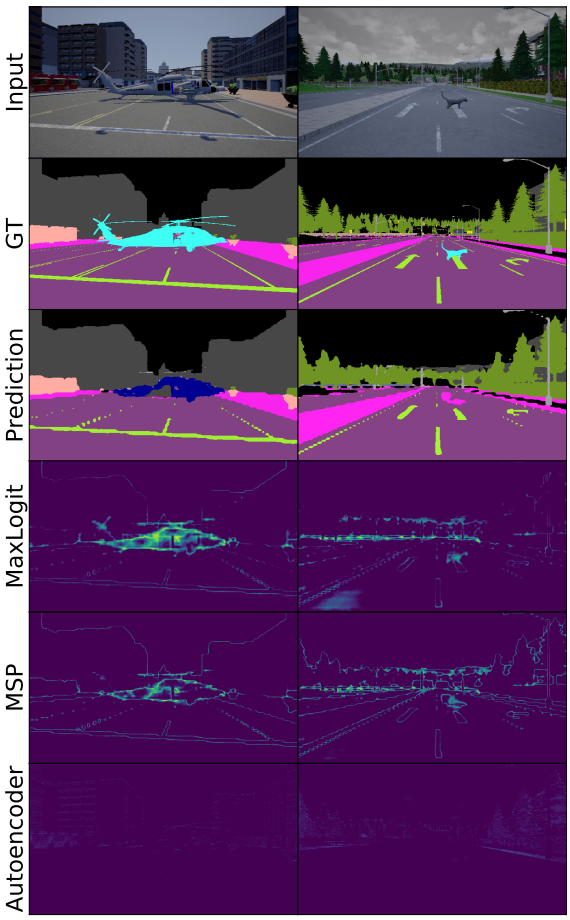


Figure 4.3: A sample of anomalous scenes, model predictions, and anomaly scores. The anomaly scores are thresholded to the top 10% of values for visualization. GT is ground truth, the autoencoder model is based on the spatial autoencoder used in (Baur et al. 2019), MSP is the maximum softmax probability baseline (Hendrycks and Gimpel 2017), and MaxLogit is the method we propose as a new baseline for large-scale settings.

dataset, we use the Unreal Engine along with the CARLA simulation environment (Dosovitskiy et al. 2017). From several months of development and testing including customization of the Unreal Engine and CARLA, we can insert foreign entities into a scene while having them be properly integrated. Unlike previous work, this avoids the issues of inconsistent chromatic aberration, edge effects, differences in environmental lighting, and other simple cues that an object is anomalous. Additionally, using a simulated environment allows us to dynamically insert diverse anomalous objects in any location and have them render properly with changes to lighting and weather including time of day, cloudy skies, and rain.

We use 3 towns from CARLA for training, from which we collect RGB images and their

respective semantic segmentation maps to serve as our training data for our semantic segmentation model. We generate a validation set from the fourth town. Finally, we reserve the fifth and sixth town as our test set. We insert anomalies taken from the Digimation Model Bank Library and semantic ShapeNet (ShapeNetSem) (Savva, Chang, and Hanrahan 2015) into the test set in order to evaluate methods for out-of-distribution detection. In total, we use 250 unique anomaly models of diverse types. There are 12 classes used for training: background, road, street lines, traffic signs, sidewalk, pedestrian, vehicle, building, wall, pole, fence, and vegetation. The thirteenth class is the anomaly class that is only used at test time. We collect 5,125 image and semantic segmentation ground truth pairs for training, 1,031 pairs without anomalies for validation, and 1,500 test pairs with anomalies.

The BDD-Anomaly Dataset. BDD-Anomaly is an anomaly segmentation dataset with real images in diverse conditions. We source BDD-Anomaly from BDD100K (Yu et al. 2018), a large-scale semantic segmentation dataset with diverse driving conditions. The original data consists in 7000 images for training and 1000 for validation. There are 18 original classes. We choose *motorcycle*, *train*, and *bicycle* as the anomalous object classes and remove all images with these objects from the training and validation sets. This yields 6,280 training pairs, 910 validation pairs without anomalies, and 810 testing pairs with anomalous objects.

4.4.1 Experiments

Metrics. To evaluate out-of-distribution detectors in large-scale settings, we use three standard metrics of detection performance: area under the ROC curve (AUROC), false positive rate at 95% recall (FPR95), and area under the precision-recall curve (AUPR). The AUROC and AUPR are important metrics, because they give a holistic measure of performance when the cutoff for detecting anomalies is not a priori obvious or when we want to represent the performance of a detection method across several different cutoffs.

The AUROC can be thought of as the probability that an anomalous example is given a higher score than an ordinary example. Thus, a higher score is better, and an uninformative detector has

a AUROC of 50%. AUPR provides a metric more attuned to class imbalances, which is relevant in anomaly and failure detection, when the number of anomalies or failures may be relatively small. Last, the FPR95 metric consists of measuring the false positive rate at 95%. This is important because it tells us how many false positives (i.e. false alarms) are necessary for a given method to achieve a desired recall. This desired recall may be thought of as a safety threshold. Moreover, anomalies and system failures may require human intervention, so a detector requiring little human intervention while still detecting most anomalies is of pecuniary importance.

In anomaly segmentation experiments, each pixel is treated as a prediction, resulting in many predictions to evaluate. To fit these in memory, we compute the metrics on each image and average over the images to obtain final values.

Methods. Our first baseline is pixel-wise Maximum Softmax Probability (MSP). Introduced in Hendrycks and Gimpel 2017 for multi-class out-of-distribution detection, we directly port this baseline to anomaly segmentation. Alternatively, the background class might serve as an anomaly detector, because it contains everything not in the other classes. To test this hypothesis, "Background" uses the posterior probability of the background class as the anomaly score. The Dropout method leverages MC Dropout (Gal and Ghahramani 2016) to obtain an epistemic uncertainty estimate. We follow the implementation by Kendall, Badrinarayanan, and Cipolla 2015.MC Dropout is computed by leaving dropout on during inference and then one runs several forward passes of the image through the network. This creates a set of predictions for the given object that we use to compute the variance over the set of predictions. The variance over the set of predictions serves as the anomaly score. A higher variance implies higher uncertainty about which class the object belongs to. We also experiment with an autoencoder baseline similar to Baur et al. 2019; Haselmann, Gruber, and Tabatabai 2018 where pixel-wise reconstruction loss is used as the anomaly score. By this we mean that we run the image through the autoencoder and subtract the resulting image from the input. The absolute difference in magnitude serves as the anomaly score. This method is called AE. The "Branch" method is a direct port of the confidence branch detector from DeVries and Taylor 2018 to pixel-wise prediction. This method trains a

separate final output score along with a classification output. The output score is multiplied to the predictions so that the predictions are scaled based on the model's confidence. The training involves the slight modification to cross entropy $\mathcal{L} = -ylog(p \cdot c)$ where y is the label, p is the probability, and c is the confidence score. The confidence score is always between 0-1 by applying the sigmoid function ensuring that the cross entropy function is still valid. The confidence is trained via backpropagation similarly to the classification prediction. Finally, we use the MaxLogit method described in earlier sections.

For all of the baselines except the autoencoder, we train a PSPNet (Zhao et al. 2017) decoder with a ResNet-101 encoder (He et al. 2015) for 20 epochs. The PSPNet follows a similar pattern to Zoomout (Mostajabi, Yadollahpour, and Shakhnarovich 2015) or Hypercolumns (Hariharan et al. 2014) whereby the activations of the convolutional layers are concatenated together before finally undergoing a fully connected layer to arrive at the appropriate dimensionality. We train both the encoder and decoder using SGD with momentum of 0.9, a learning rate of 2×10^{-2}, and learning rate decay of 10^{-4}. For the autoencoder, we use a 4-layer U-Net (Ronneberger, Fischer, and Brox 2015) with a spatial latent code as in (Baur et al. 2019). The U-Net also uses batch norm and is trained for 10 epochs.

To evaluate the methods, we take all of the scores per pixel that belong to anomalies and all of the scores for the remaining pixels. Then we sort the scores, thereby allowing us to compute the false positive rate and true positive rate at every threshold. We use the thresholds to compute the AUROC giving us the probability that we correctly select an in-distribution pixel with the method we're evaluating. We also predefine a set threshold at 5% and compute the false positive rate for that threshold. Due to the large number of scores to be evaluated and sorted, we take the mean over all the images of each evaluation as a final report.

Results and Analysis. MaxLogit outperforms all other methods across the board by a substantial margin. The intuitive baseline of using the posterior for the background class to detect anomalies performs poorly, which suggests that the background class may not align with rare visual features. Even though reconstruction-based scores succeed in product fault segmentation,

		MSP	Branch	Background	Dropout	AE	MaxLogit
StreetHazards	FPR95 ↓	33.7	68.4	69.0	79.4	91.7	**26.5**
	AUROC ↑	87.7	65.7	58.6	69.9	66.1	**89.3**
	AUPR ↑	6.6	1.5	4.5	7.5	2.2	**10.6**
BDD-Anomaly	FPR95 ↓	24.5	25.6	40.1	16.6	74.1	**14.0**
	AUROC ↑	87.7	85.6	69.7	90.8	64.0	**92.6**
	AUPR ↑	3.7	3.9	1.1	4.3	0.7	**5.4**

Table 4.3: Results on the Combined Anomalous Object Segmentation benchmark. AUPR is low across the board due to the large class imbalance, but all methods perform substantially better than chance. MaxLogit obtains the best performance. All results are percentages.

we find that the AE method performs poorly on the CAOS benchmark, which may be due to the more complex domain. AUPR for all methods is low, indicating that the large class imbalance presents a serious challenge. However, the substantial improvements with the MaxLogit method suggest that progress on this task is possible and there is much room for improvement.

In figure 4.3, we can qualitatively see that both MaxLogit and MSP have a high number of false positives, as they assign high anomaly scores to semantic boundaries, a problem also observed in the recent works of Blum et al. 2019; Angus 2019. However, the problem is less severe in MaxLogit. A potential explanation for this could be due to two effects. The first we mentioned earlier in the benefits of MaxLogit over MSP (see section 4.3) in that the inter-class variance is better preserved in MaxLogit over MSP. The second effect builds off of the first, by having a greater range in MaxLogit as compared to MSP bilinear upsampling from the models final output to the final output image creates much sharper boundaries. This is because the interpolation of points that are already close (with MSP) will blur the boundaries much more and cause more pixels to become classified as in-distribution by exceeding the threshold.

Autoencoder-based methods are qualitatively different from approaches using the softmax probabilities, because they model the input itself and can avoid boundary effects seen in the MaxLogit and MSP rows of figure 4.3. While autoencoder methods are successful in medical anomaly segmentation and product fault detection, we find the AE baseline to be ineffective in the more complex domain of street scenes. The last row of figure 4.3 shows pixel-wise reconstruction loss on example images from StreetHazards. Anomalies are not distinguished well from in-

distribution elements of the scene. New methods must be developed to mitigate the boundary effects faced by softmax-based methods while also attaining good detection performance.

4.5 Conclusion

We scaled out-of-distribution detection to more realistic, large-scale settings by developing a novel benchmark for OOD segmentation. The CAOS benchmark for anomaly segmentation consists of diverse, naturally-integrated anomalous objects in driving scenes. Baseline methods on the CAOS benchmark substantially improve on random guessing but are still lacking, indicating potential for future work. We also investigated using multi-label classifiers for out-of-distribution detection and established an experimental setup for this previously unexplored setting. On large-scale multi-class image datasets, we identified an issue faced by existing baselines and proposed the maximum logit detector as a natural solution. Interestingly, this detector also provides consistent and significant gains in the multi-label and anomaly segmentation settings, thereby establishing it as a new baseline in place of the maximum softmax probability baseline on large-scale OOD detection problems. In all, we hope that our simple baseline and our new OOD segmentation benchmark will enable further research on out-of-distribution detection for real-world safety-critical environments.

CHAPTER 5

NEURAL AUGMENTATIONS

5.1 Overview

While the research community must create robust models that generalize to new scenarios, the robustness literature (Dodge and Karam 2017; Geirhos et al. 2020b) lacks consensus on evaluation benchmarks and contains many dissonant hypotheses. While Hendrycks, Liu, et al. 2020 find that many recent language models are already robust to many forms of distribution shift, Yin et al. 2019, and Geirhos et al. 2019 find that vision models are largely fragile and argue that data augmentation offers a solution. In contrast, Taori et al. 2020b provide results suggesting that using pretraining and improving in-distribution test set accuracy improve natural robustness, whereas other methods do not.

In this chapter we articulate and systematically study seven robustness hypotheses. The first four hypotheses concern *methods* for improving robustness, while the last three hypotheses concern abstract *properties* about robustness. These hypotheses are as follows.

- *Larger Models*: increasing model size improves robustness (Xie and Yuille 2020).
- *Self-Attention*: adding self-attention layers to models improves robustness (Hendrycks et al. 2019).
- *Diverse Data Augmentation*: robustness can increase through data augmentation (Yin et al. 2019).
- *Pretraining*: pretraining on larger and more diverse datasets improves robustness (Orhan 2019; Hendrycks, Lee, and Mazeika 2019).
- *Texture Bias*: convolutional networks are biased towards texture, which harms robustness (Geirhos et al. 2019).
- *Only IID Accuracy Matters*: accuracy on independent and identically distributed test data entirely determines natural robustness (Taori et al. 2020a).
- *Synthetic* $\not\Longrightarrow$ *Natural*: *synthetic* robustness interventions including diverse data augmentations do not help with robustness on *naturally occurring* distribution shifts (Taori et al. 2020b).

Figure 5.1: Images from our three new datasets ImageNet-Renditions (ImageNet-R), DeepFashion Remixed (DFR), and StreetView StoreFronts (SVSF). The SVSF images are recreated from the public Google StreetView, copyright Google 2020. Our datasets test robustness to various naturally occurring distribution shifts including rendition style, camera viewpoint, and geography.

It has been difficult to arbitrate these hypotheses because existing robustness datasets preclude the possibility of controlled experiments by varying multiple aspects simultaneously. For instance, *Texture Bias* was initially investigated with synthetic distortions (Geirhos et al. 2018), which conflicts with the *Synthetic* $\not\Rightarrow$ *Natural* hypothesis. On the other hand, natural distribution shifts often affect many factors (e.g., time, camera, location, etc.) simultaneously in unknown ways (Recht et al. 2019; Hendrycks et al. 2019). Existing datasets also lack diversity such that it is hard to extrapolate which methods will improve robustness more broadly. To address these issues and test the seven hypotheses outlined above, we introduce three new robustness benchmarks and a new data augmentation method.

First we introduce ImageNet-Renditions (ImageNet-R), a 30,000 image test set containing various renditions (e.g., paintings, embroidery, etc.) of ImageNet object classes. These renditions are naturally occurring, with textures and local image statistics unlike those of ImageNet images, allowing us to more cleanly separate the *Texture Bias* and *Synthetic* $\not\Rightarrow$ *Natural* hypotheses.

Next, we investigate natural shifts in the image capture process with StreetView StoreFronts (SVSF) and DeepFashion Remixed (DFR). SVSF contains business storefront images taken from Google Streetview, along with metadata allowing us to vary location, year, and even the camera

type. DFR leverages the metadata from DeepFashion2 (Ge et al. 2019) to systematically shift object occlusion, orientation, zoom, and scale at test time. Both SVSF and DFR provide distribution shift controls and do not alter texture, which remove possible confounding variables affecting prior benchmarks. DFR is discussed in greater detail in 6.4.

Finally, we contribute DeepAugment to increase robustness to some new types of distribution shift. This augmentation technique uses image-to-image neural networks for data augmentation, not data-independent Euclidean augmentations like image shearing or rotating as in previous work. DeepAugment achieves state-of-the-art robustness on our newly introduced ImageNet-R benchmark and a corruption robustness benchmark. DeepAugment can also be combined with other augmentation methods to outperform a model pretrained on $1000\times$ more labeled data.

After examining our results on these three datasets and others, we can rule out several of the above hypotheses while strengthening support for others. As one example, we find that synthetic data augmentation robustness interventions improve accuracy on ImageNet-R and real-world image blur distribution shifts, providing clear counterexamples to *Synthetic* $\not\Longrightarrow$ *Natural* while lending support to the *Diverse Data Augmentation* and *Texture Bias* hypotheses. In the conclusion, we summarize the various strands of evidence for and against each hypothesis. Across our many experiments, we do not find a general method that consistently improves robustness, and some hypotheses require additional qualifications. While robustness is often spoken of and measured as a single scalar property like accuracy, our investigations suggest that robustness is not so simple. In light of our results, we hypothesize in the conclusion that robustness is *multivariate*.

5.2 Related Work

Robustness Benchmarks. Recent works (Hendrycks and Dietterich 2019; Recht et al. 2019; Hendrycks, Liu, et al. 2020) have begun to characterize model performance on out-of-distribution (OOD) data with various new test sets, with dissonant findings. For instance, Hendrycks, Liu, et al. 2020 demonstrate that modern language processing models are moderately robust to numerous naturally occurring distribution shifts, and that *Only IID Accuracy Matters* is inaccurate for natural

language tasks. For image recognition, Hendrycks and Dietterich 2019 analyze image models and show that they are sensitive to various simulated image corruptions (e.g., noise, blur, weather, JPEG compression, etc.) from their "ImageNet-C" benchmark.

Recht et al. 2019 reproduce the ImageNet (Russakovsky et al. 2014) validation set for use as a benchmark of naturally occurring distribution shift in computer vision. Their evaluations show a 11-14% drop in accuracy from ImageNet to the new validation set, named ImageNetV2, across a wide range of architectures. Taori et al. 2020b use ImageNetV2 to measure natural robustness and dismiss *Diverse Data Augmentation*. Engstrom et al. 2020 identify statistical biases in ImageNetV2's construction, and they estimate that reweighting ImageNetV2 to correct for these biases results in a less substantial 3.6% drop.

In contrast to *adversarial* robustness (Szegedy et al. 2013; Ian J Goodfellow, Shlens, and Szegedy 2014), we focus instead on robustness to unconstrained out-of-distribution data measured on non-interactive benchmarks. (Carlini et al. 2019) detail the inherent practical difficulty in evaluating adversarial robustness, and (Gilmer et al. 2018) argue for the inclusion of unconstrained input modifications in the threat model of attacks against machine learning systems.

Data Augmentation. Geirhos et al. 2019; Yin et al. 2019; Hendrycks, Mu, et al. 2020 demonstrate that data augmentation can improve robustness on ImageNet-C. The space of augmentations that help robustness includes various types of noise (Aleksander Madry et al. 2018; Rusak et al. 2020; Lopes et al. 2019), highly unnatural image transformations (Geirhos et al. 2019; Yun et al. 2019; Hongyi Zhang et al. 2017), or compositions of simple image transformations such as Python Imaging Library operations (Cubuk et al. 2018; Hendrycks, Mu, et al. 2020). Some of these augmentations can improve accuracy on in-distribution examples as well as on out-of-distribution (OOD) examples.

Transfer learning Pretraining models on larger datasets has been been demonstrated to improve robustness on ImageNet-C. Orhan 2019 report that models trained on the JFT-300m dataset (Sun et al. 2017) and the weakly-supervised Instagram dataset (Mahajan et al. 2018) improve

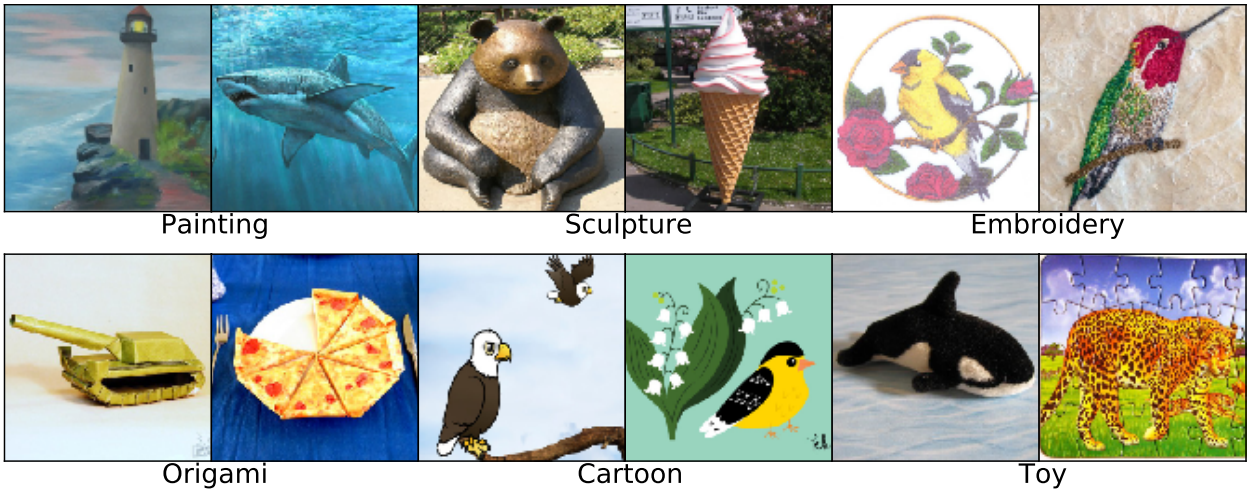


Figure 5.2: ImageNet-Renditions (ImageNet-R) contains 30,000 images of ImageNet objects with different textures and styles. This figure shows only a portion of ImageNet-R's numerous rendition styles. The rendition styles (e.g., "Toy") are for clarity and are *not* ImageNet-R's classes; ImageNet-R's classes are a subset of 200 ImageNet classes. ImageNet-R emphasizes shape over texture.

robustness on the ImageNet-C by significant margins.

5.3 New Benchmarks

In order to evaluate the seven robustness hypotheses, we introduce three new benchmarks that capture new types of naturally occurring distribution shifts. ImageNet-Renditions (ImageNet-R) is a newly collected test set intended for ImageNet classifiers, whereas StreetView StoreFronts (SVSF) and DeepFashion Remixed (DFR) each contain their own training sets and multiple test sets. SVSF and DFR split data into a training and test sets based on various image attributes stored in the metadata. For example, we can select a test set with images produced by a camera different from the training set camera. We now describe the structure and collection of each dataset.

5.3.1 ImageNet-Renditions (ImageNet-R)

While current classifiers can learn some aspects of an object's shape (Mordvintsev, Olah, and Tyka 2015), they nonetheless rely heavily on natural textural cues (Geirhos et al. 2019). In contrast, human vision can process abstract visual renditions. For example, humans can recognize visual

scenes from line drawings as quickly and accurately as they can from photographs (Biederman and Ju 1988). Even some primates species have demonstrated the ability to recognize shape through line drawings (Itakura 1994; Tanaka 2006).

To measure generalization to various abstract visual renditions, we create the ImageNet-Rendition (ImageNet-R) dataset. ImageNet-R contains various artistic renditions of object classes from the original ImageNet dataset. Note the original ImageNet dataset discouraged such images since annotators were instructed to collect "photos only, no painting, no drawings, etc." (Deng 2012). We do the opposite.

Data Collection. ImageNet-R contains 30,000 image renditions for 200 ImageNet classes. We collect images primarily from Flickr and use queries such as "art," "cartoons," "graffiti," "embroidery," "graphics," "origami," "paintings," "patterns," "plastic objects," "plush objects," "sculptures," "line drawings," "tattoos," "toys," "video game," and so on. Examples are depicted in figure 5.2. Images are filtered by Amazon MTurk workers using a modified collection interface from ImageNetV2 (Recht et al. 2019). The resulting images are then manually filtered by graduate students. ImageNet-R also includes the line drawings from (H. Wang et al. 2019), excluding horizontally mirrored duplicate images, pitch black images, and images from the incorrectly collected "pirate ship" class.

5.3.2 StreetView StoreFronts (SVSF)

Computer vision applications often rely on data from complex pipelines that span different hardware, times, and geographies. Ambient variations in this pipeline may result in unexpected performance degradation, such as degradations experienced by health care providers in Thailand deploying laboratory-tuned diabetic retinopathy classifiers in the field (Beede et al. 2020). In order to study the effects of shifts in the image capture process we collect the StreetView StoreFronts (SVSF) dataset, a new image classification dataset sampled from Google StreetView imagery (Anguelov et al. 2010) focusing on three distribution shift sources: country, year, and camera.

Data Collection. SVSF consists of cropped images of business store fronts extracted from StreetView images by an object detection model. Each store front image is assigned the class label of the associated Google Maps business listing through a combination of machine learning models and human annotators. We combine several visually similar business types (e.g. drugstores and pharmacies) for a total of 20 classes, listed Appendix A.6. We are currently unable to release the SVSF data publicly.

Splitting the data along the three metadata attributes of country, year, and camera, we create one training set and five test sets. We sample a training set and an in-distribution test set (200K and 10K images, respectively) from images taken in US/Mexico/Canada during 2019 using a "new" camera system. We then sample four OOD test sets (10K images each) which alter one attribute at a time while keeping the other two attributes consistent with the training distribution. Our test sets are year: 2017, 2018; country: France; and camera: “old.”

5.4 DeepAugment

In order to further explore the *Diverse Data Augmentation* hypothesis, we introduce a new data augmentation technique we call DeepAugment. DeepAugment works by passing an image through an image-to-image network (such as an image autoencoder or a superresolution network), but rather than processing the image normally, we distort the internal weights and activations. We distort the image-to-image network’s weights by applying randomly sampled operations such as zeroing, negating, convolving, transposing, applying activation functions, and so on. This creates diverse but semantically consistent images as illustrated in figure 5.3. We provide the pseudocode in Appendix A.7. Whereas most previous data augmentations techniques use simple augmentation primitives applied to the raw image itself, we stochastically distort the internal representations of image-to-image networks to augment images. We did not experiment with any geometric image-to-image networks such as NeRF (Mildenhall et al. 2020). We have results in table 5.1 that shows how DeepAugment works well with Augmix which does incorporate geometric transformations.

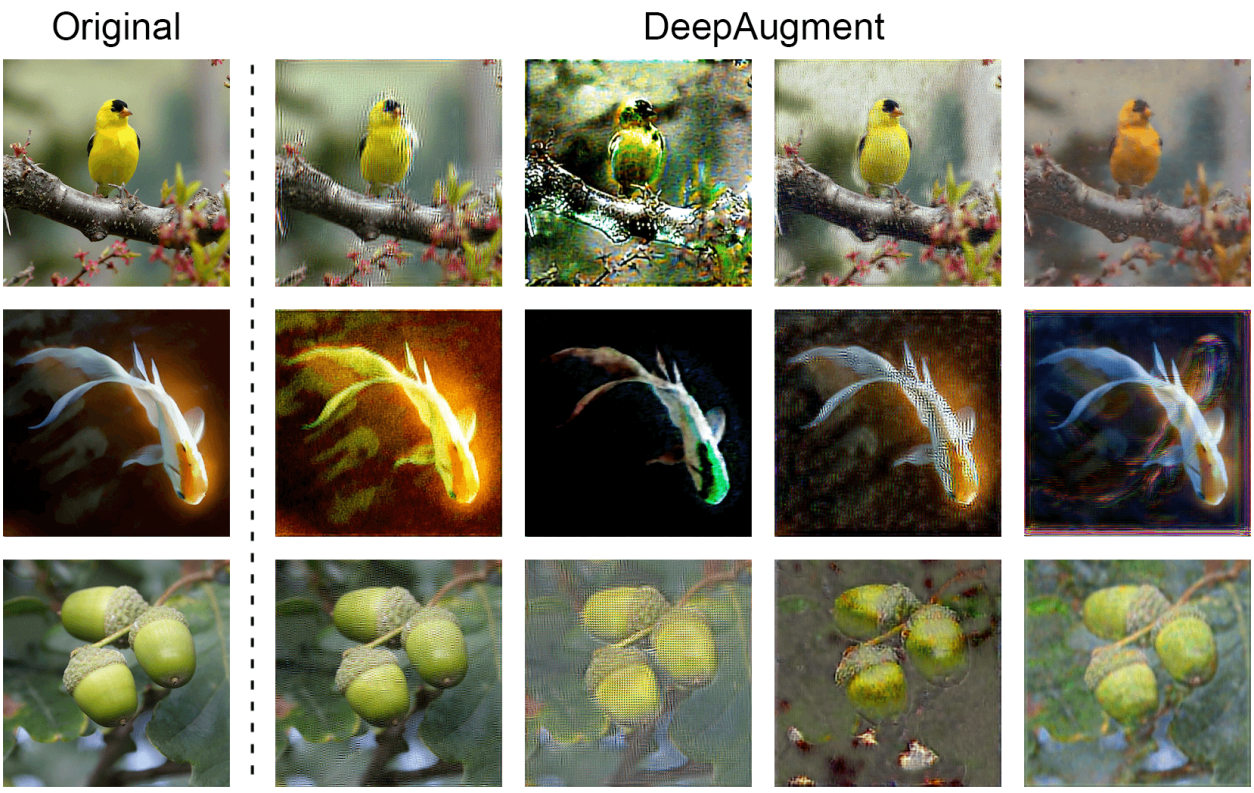


Figure 5.3: DeepAugment examples preserve semantics, are data-dependent, and are far more visually diverse than augmentations such as rotations.

5.5 Experiments

5.5.1 Setup

In this section we briefly describe the evaluated models, pretraining techniques, self-attention mechanisms, data augmentation methods, and note various implementation details.

Model Architectures and Sizes. Most experiments are evaluated on a standard ResNet-50 model (He et al. 2015). Model size evaluations use ResNets or ResNeXts (Xie et al. 2016) of varying sizes.

Pretraining. For pretraining we use ImageNet-21K which contains approximately 21,000 classes and approximately 14 million labeled training images, or around $10\times$ more labeled training data than ImageNet-1K. We tune Kolesnikov et al. 2019's ImageNet-21K model. We also use a large pre-trained ResNeXt-101 model from Mahajan et al. 2018. This was pre-trained on on approximately

1 billion Instagram images with hashtag labels and fine-tuned on ImageNet-1K. This Weakly Supervised Learning (WSL) pretraining strategy uses approximately $1000\times$ more labeled data.

Self-Attention. When studying self-attention, we employ CBAM (Woo et al. 2018) and SE (Hu, Shen, and Sun 2018) modules, two forms of self-attention that help models learn spatially distant dependencies.

Data Augmentation. We use Style Transfer, AugMix, and DeepAugment to analyze the *Diverse Data Augmentation* hypothesis, and we contrast their performance with simpler noise augmentations such as Speckle Noise and adversarial noise. Style transfer (Geirhos et al. 2019) uses a style transfer network to apply artwork styles to training images. AugMix (Hendrycks, Mu, et al. 2020) randomly composes simple augmentation operations (e.g., translate, posterize, solarize). DeepAugment, introduced above, distorts the weights and feedforward passes of image-to-image models to generate image augmentations. Speckle Noise data augmentation muliplies each pixel by $(1 + x)$ with x sampled from a normal distribution (Rusak et al. 2020; Hendrycks and Dietterich 2019). We also consider adversarial training as a form of adaptive data augmentation and use the model from (Wong, Rice, and Kolter 2020) trained against ℓ_∞ perturbations of size $\varepsilon = 4/255$.

5.5.2 Results

We now perform experiments on ImageNet-R, and StreetView StoreFronts leaving results on DeepFashion Remixed to Chapter 6 on multilabel OOD. We also evaluate on ImageNet-C and compare and contrast it with real distribution shifts.

ImageNet-R. Table 5.1 shows performance on ImageNet-R as well as on ImageNet-200 (the original ImageNet data restricted to ImageNet-R's 200 classes). This has several implications regarding the four method-specific hypotheses. *Pretraining* with ImageNet-21K (approximately $10\times$ labeled data) hardly helps. Appendix A.5 shows WSL pretraining can help, but Instagram has renditions, while ImageNet excludes them; hence we conclude comparable pretraining was ineffective. Notice *Self-Attention* increases the IID/OOD gap. Compared to simpler data augmentation techniques

Error Rates	ImageNet-200 (%)	ImageNet-R (%)	Gap
ResNet-50	7.9	63.9	56.0
+ ImageNet-21K *Pretraining* ($10\times$ labeled data)	7.0	62.8	55.8
+ CBAM (*Self-Attention*)	7.0	63.2	56.2
+ ℓ_∞ Adversarial Training	25.1	68.6	43.5
+ Speckle Noise	8.1	62.1	54.0
+ Style Transfer Augmentation	8.9	58.5	49.6
+ AugMix	7.1	58.9	51.8
+ DeepAugment	7.5	57.8	50.3
+ DeepAugment + AugMix	8.0	53.2	45.2
ResNet-152 (*Larger Models*)	6.8	58.7	51.9

Table 5.1: ImageNet-200 and ImageNet-R top-1 error rates. ImageNet-200 uses the same 200 classes as ImageNet-R. DeepAugment+AugMix improves over the baseline by over 10 percentage points. ImageNet-21K Pretraining tests *Pretraining* and CBAM tests *Self-Attention*. Style Transfer, AugMix, and DeepAugment test *Diverse Data Augmentation* in contrast to simpler noise augmentations such as ℓ_∞ Adversarial Noise and Speckle Noise. While there remains much room for improvement, results indicate that progress on ImageNet-R is tractable.

such as Speckle Noise, the *Diverse Data Augmentation* techniques of Style Transfer, AugMix, and DeepAugment improve generalization. Note AugMix and DeepAugment improve in-distribution performance whereas Style transfer hurts it. Also, our new DeepAugment technique is the best standalone method with an error rate of 57.8%. Last, *Larger Models* reduce the IID/OOD gap. Full results for all evaluated models can be found in the Appendix in table A.4.

Regarding the three more abstract hypotheses, biasing networks away from natural textures through diverse data augmentation improved performance, so we find support for the *Texture Bias* hypothesis. The IID/OOD generalization gap varies greatly which condtradicts *Only IID Accuracy Matters*. Finally, since ImageNet-R contains real-world examples, and since synthetic data augmentation helps on ImageNet-R, we now have clear evidence against the *Synthetic* $\not\Longrightarrow$ *Natural* hypothesis.

StreetView StoreFronts. In table 5.2, we evaluate data augmentation methods on SVSF and find that all of the tested methods have mostly similar performance and that no method helps much on country shift, where error rates roughly double across the board. Images captured in France contain noticeably different architectural styles and storefront designs than those captured

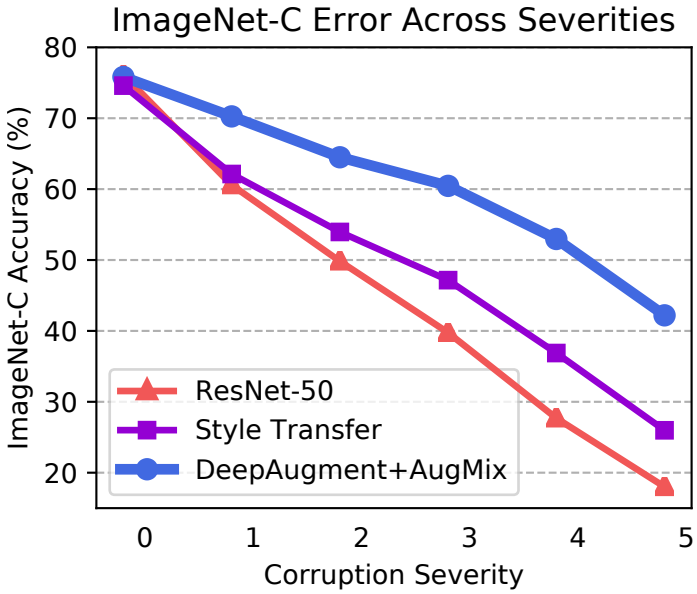


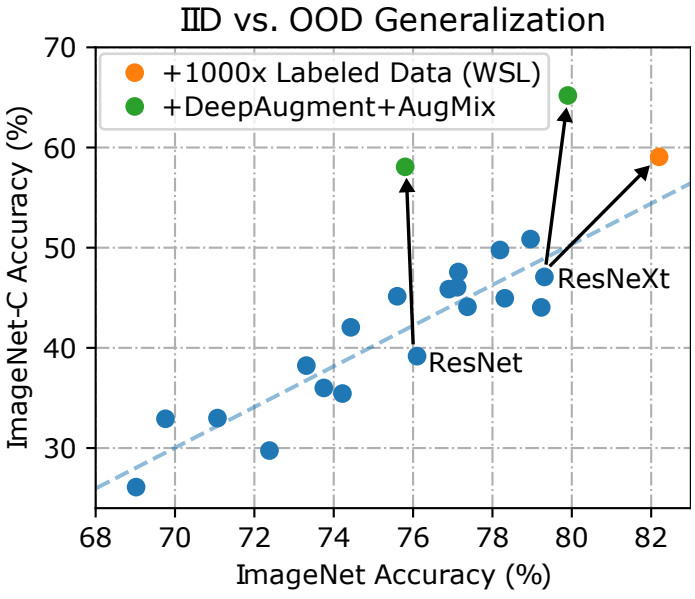
Figure 5.4: Accuracy as a function of corruption severity. Severity "0" denotes clean data. Data augmentation methods such as DeepAugment with AugMix shift the entire Pareto frontier outward.

in US/Mexico/Canada; meanwhile, we are unable to find conspicuous and consistent indicators of the camera and year. This may explain the relative insensitivity of evaluated methods to the camera and year shifts. Overall *Diverse Data Augmentation* shows limited benefit, suggesting either that data augmentation primarily helps combat texture bias as with ImageNet-R, or that existing augmentations are not diverse enough to capture high-level semantic shifts such as building architecture.

		Hardware	Year		Location
Network	IID	Old	2017	2018	France
ResNet-50	27.2	28.6	27.7	28.3	56.7
+ Speckle Noise	28.5	29.5	29.2	29.5	57.4
+ Style Transfer	29.9	31.3	30.2	31.2	59.3
+ DeepAugment	30.5	31.2	30.2	31.3	59.1
+ AugMix	26.6	28.0	26.5	27.7	55.4

Table 5.2: SVSF classification error rates. Networks are robust to some natural distribution shifts but are substantially more sensitive the geographic shift. Here *Diverse Data Augmentation* hardly helps.

ImageNet-C. We now consider a previous robustness benchmark to reassess all seven hypothe-



htbp

Figure 5.5: ImageNet accuracy and ImageNet-C accuracy. Previous architectural advances slowly translate to ImageNet-C performance improvements, but DeepAugment+AugMix on a ResNet-50 yields a $\approx 19\%$ accuracy increase.

ses. We use the ImageNet-C dataset (Hendrycks and Dietterich 2019) which applies 15 common image corruptions (e.g., Gaussian noise, defocus blur, simulated fog, JPEG compression, etc.) across 5 severities to ImageNet-1K validation images. We find that DeepAugment improves robustness on ImageNet-C. Figure 5.5 shows that when models are trained with AugMix and DeepAugment, they attain the state-of-the-art, break the trendline, and exceed the corruption robustness provided by training on $1000\times$ more labeled training data. Note the augmentations from AugMix and DeepAugment are disjoint from ImageNet-C's corruptions. Full results are shown in Appendix A.5's A.5. This is evidence against the *Only IID Accuracy Matters* hypothesis and is evidence for the *Larger Models*, *Self-Attention*, *Diverse Data Augmentation*, *Pretraining*, and *Texture Bias* hypotheses.

Taori et al. 2020b remind us that ImageNet-C uses various *synthetic* corruptions and suggest that they are divorced from real-world robustness. Real-world robustness requires generalizing to naturally occurring corruptions such as snow, fog, blur, low-lighting noise, and so on, but it is an open question whether ImageNet-C's simulated corruptions meaningfully approximate real-world corruptions.

Hypothesis	ImageNet-C	Real Blurry Images	ImageNet-R	DFR	SVSF
Larger Models	+	+	+	−	
Self-Attention	+	+	−	−	
Diverse Data Augmentation	+	+	+	−	−
Pretraining	+	+	−	−	

Table 5.3: A highly simplified account of each hypothesis when tested against different datasets. Evidence for is denoted "+", and "−" denotes an absence of evidence or evidence against.

We collect a small dataset of 1,000 real-world blurry images and find that ImageNet-C can track robustness to real-world corruptions. We collect the "Real Blurry Image" dataset with Flickr and query ImageNet object class names concatenated with the word "blurry." We then evaluate various models on real-world blurry images and find that *all* the robustness interventions that help with ImageNet-C also help with real-world blurry images. Hence ImageNet-C can track performance on real-world corruptions. Moreover, DeepAugment+AugMix has the lowest error rate on Real Blurry Images, which again contradicts the *Synthetic* $\not\Longrightarrow$ *Natural* hypothesis. Appendix A.4 has full results. The upshot is that ImageNet-C is a controlled and systematic proxy for real-world robustness.

5.6 Conclusion

We introduced two new multi-class benchmarks, ImageNet-Renditions, and StreetView StoreFronts. With these benchmarks, we thoroughly tested seven robustness hypotheses–four about methods for robustness, and three about the nature of robustness.

Let us consider the first four hypotheses, using the new information from ImageNet-C and our three new benchmarks. The *Larger Models* hypothesis was supported with ImageNet-C and ImageNet-R, but not with DFR. While *Self-Attention* noticeably helped ImageNet-C, it did not help with ImageNet-R and DFR. *Diverse Data Augmentation* was ineffective for SVSF and DFR, but it greatly improved ImageNet-C and ImageNet-R accuracy. *Pretraining* greatly helped with ImageNet-C but hardly helped with ImageNet-R. This is summarized in table 5.3. It was not obvious *a priori* that synthetic *Diverse Data Augmentation* could improve ImageNet-R accuracy,

nor did previous research suggest that *Pretraining* would sometimes be ineffective. While no single method consistently helped across all distribution shifts, some helped more than others.

Our analysis of these four hypotheses have implications for the remaining three hypotheses. Regarding *Texture Bias*, ImageNet-R shows that networks do not generalize well to renditions (which have different textures), but that diverse data augmentation (which often distorts textures) can recover accuracy. More generally, larger models and diverse data augmentation consistently helped on ImageNet-R, ImageNet-C, and Blurry Images, suggesting that these two interventions reduce texture bias. However, these methods helped little for geographic shifts, showing that there is more to robustness than texture bias alone. Regarding *Only IID Accuracy Matters*, while IID accuracy is a strong predictor of OOD accuracy, it is not decisive—Table 5.3 shows that many methods improve robustness across multiple distribution shifts, and recent experiments in NLP provide further counterexamples (Hendrycks, Liu, et al. 2020). Finally, *Synthetic* $\not\Longrightarrow$ *Natural* has clear counterexamples given that DeepAugment greatly increases accuracy on ImageNet-R and Real Blurry Images. In summary, some previous hypotheses are implausible, and the Texture Bias hypothesis has the most support.

Our seven hypotheses presented several conflicting accounts of robustness. What led to this conflict? We suspect it is because robustness is not one scalar like accuracy. The research community is reasonable in judging IID accuracy with a *univariate* metric like ImageNet classification accuracy, as models with higher ImageNet accuracy reliably have better fine-tuned classification accuracy on other tasks (Kornblith, Shlens, and Le 2018). In contrast, we argue it is too simplistic to judge OOD accuracy with a univariate metric like, say, ImageNetV2 or ImageNet-C accuracy. Instead we hypothesize that robustness is multivariate. This *Multivariate* hypothesis means that there is not a single scalar model property that wholly governs natural model robustness.

If robustness has many faces, future work should evaluate robustness using many distribution shifts; for example, ImageNet models should at least be tested against ImageNet-C and ImageNet-R. Future work could further characterize the space of distribution shifts. However there are now more out-of-distribution robustness datasets than there are published robustness methods. Hence

the research community should prioritize creating new robustness methods. If our *Multivariate* hypothesis is true, multiple tests are necessary to develop models that are both robust and safe.

CHAPTER 6

MULTI-LABEL OUT-OF-DISTRIBUTION DETECTION

6.1 Overview

Research in multi-label classification has been in the shadow of multi-class classification for the greater part of a decade (Tidake and Sane 2018). This lack of focus is not entirely unjustified though, as many of the improvements in algorithm and model design have carried over to multi-label classification (He et al. 2015; Chen et al. 2019; Ben-Baruch et al. 2020). While improvements have been made there still remains a considerable gap between multi-class performance and multi-label performance.

This gap is even more pronounced when considering the difference in robustness difference from multi-class to multi-label. Previous work in robustness focused on small whole-image anomaly detection with surprisingly no research studying the multi-label setting. In the previous two chapters, we have focused on scaling up to large-scale datasets which presents unique challenges such as a plethora of fine-grained object classes, see Chatper 4 and Chapter 5. In this chapter, we demonstrate that the maximum softmax probability (MSP) detector, a state-of-the-art method for small-scale problems, does not scale well to these challenging conditions. Moreover, in the multi-label setting the MSP detector cannot naturally be applied in the first place, as it requires softmax probabilities.

Due to the limitations of the MSP, we modified it for use in the multi-label setting to the maximum logit which we covered in greater detail in Chapter 4. We also introduce a new technique that takes into account the correlations among the labels which is able to achieve comparable performance.

6.2 Related Work

Natural images often contain many objects of interest with complex relationships of co-occurrence. Multi-label image classification acknowledges this more realistic setting by allowing each image to have multiple overlapping labels. This problem has long been of interest (Everingham et al. 2009), and recent web-scale multi-label datasets demonstrate its growing importance, including Tencent ML-Images (Wu et al. 2019) and Open Images (Kuznetsova et al. 2018).

There are a few distinct techniques that have made progress in the multi-label task beyond architectural improvements. One of the earlier techniques is combining recurrent neural networks with convolutional neural networks (CNNs) (Nam et al. 2017). The output of the CNN is fed into the RNN as a sequence to sequence task akin to language translation. Most recently graph neural networks (GNNs) have been used after the output of a CNN to learn the label dependencies (Chen et al. 2019). Others have expanded on utilizing GNNs by combining them with word embeddings (Ya Wang et al. 2019).

Prior work addresses multi-label classification in various ways, such as by leveraging label dependencies (J. Wang et al. 2016). While current work on out-of-distribution detection solely considers multi-class or unsupervised settings. Yet as classifiers learn to classify more objects and process larger images, the multi-label formulation becomes increasingly natural. To our knowledge, this problem setting has yet to be explored. We provide a baselines and evaluation setup.

6.3 Methods

Datasets. For multi-label classification we use the datasets PASCAL VOC (Everingham et al. 2009), MS-COCO (Lin et al. 2014), and DeepFashion Remixed (DFR) (Ge et al. 2019). Specifically for MS-COCO and PASCAL VOC, we evaluate the models trained on these datasets, by using 20 out-of-distribution classes from ImageNet-22K. These classes have no overlap with ImageNet-1K, PASCAL VOC, or MS-COCO. The 20 classes are chosen not to overlap with ImageNet-1K since

the multi-label classifiers models are pre-trained on ImageNet-1K. We list the class WordNet IDs in the Supplementary Materials A.6.

In PASCAL VOC and MS-COCO both datasets are subject to changes in day-to-day camera operation which can cause shifts in attributes such as object size, object occlusion, camera viewpoint, and camera zoom. To measure this effect, we repurpose DeepFashion2 (Ge et al. 2019) to create the DeepFashion Remixed (DFR) dataset. We designate a training set with 48K images and create eight out-of-distribution test sets to measure performance under shifts in object size, object occlusion, camera viewpoint, and camera zoom-in. DeepFashion Remixed is a multi-label classification task since images may contain more than one clothing item per image. In this way we can control for changes in those attributes.

Architecture. For our experiments we use a ResNet-101 backbone architecture pre-trained on ImageNet-1K. We replace the final layer with a fully connected layers and apply the logistic sigmoid function for multi-label prediction.

$$\mathcal{L} = \sum_i (y_i - ln(sigmoid((logits)_i)) + (1 - y_i) \cdot ln(1 - sigmoid((logits)_i))$$

Where y_i is a d-dimensional vector of 0,1 where 1 corresponds to the presence of that class. Note that it is not a one-hot vector so the entire vector can be all ones potentially. We train each model for 50 epochs using the Adam optimizer (Kingma and Ba 2014) with hyperparameter values 10^{-4} and 10^{-5} for β_1 and β_2 respectively. For data augmentation we use standard resizing, random crops, and random flips to obtain images of size $256 \times 256 \times 3$. As a result of this training procedure, the mAP of the ResNet-101 on PASCAL VOC is 89.11% and 72.0% for MS-COCO.

For experiments on DFR data augmentation includes: a crop of random size in the (0.5 to 2.0) of the original size and a random aspect ratio of $3/4$ to $4/3$ of the original aspect ratio, which is finally resized to create a 256×256 image. For data augmentation we randomly horizontally flip the image with probability 0.5.

Detection Methods. We evaluate the trained MS-COCO and PASCAL VOC models using four different detectors described below. Even though the models are multi-label detectors because we

are feeding in single class images from Imagenet-22K we should expect all of the logits from the network to be low or zero. For the descriptions of the dectectors below the “logits” refers to the aggregate vector composed of the prediction for each class. Results are in 6.1.

- MaxLogit denotes taking the negative of the maximum value of a logits vector as the anomaly score. The logits are formed by combining all the scores from each class taken from the last layer of a neural network.

- LogitAvg is the negative of the average of the logits values taken from the last layer of the neural network.

- Isolation Forest (Liu, Ting, and Zhou 2008), denoted by IForest, works by randomly partitioning the input space into half spaces to form a decision tree. IForest needs a “training step” or setup phase before it can be used. More specifically the algorithm is as follows:

  Step 1) select a feature to split on.
  Step 2) choose a random split between min and max range for feature.
  Step 3) repeat steps 1 and 2 until all elements are singletons.
  Step 4) Repeat steps 1-3 to construct a new tree.

  The isolation score is evaluated based on the average distance to reach a terminal leaf from the trees in the ensemble. We train our isolation forest using in-distribution validation data. Note that to train the isolation forest ground truth labels of the images are not required only the knowledge that they are in-distribution. The Isolation forest can thus be considered an unsupervised learning algorithm as it does not use the image labels. Finally, we tried two approaches to construct our space used for the isolation forest. The first approach consisted of the aggregated logits vectors and the second approach consisted of using the maximum logit value. So in the first approach the space is a d-dimensional space where d equals the number of classes. The second approach consists of a 1-dimensional line based on the maximum logit possible from each image. See the MaxLogit definition 6.3. We found

that the second approach worked better and thus used that as our reported IForest values. We use the default number of trees from (Pedregosa et al. 2011) which is that of 100 trees for the ensemble.

Mathematically this works out to

$$S(x_i) = -0.5 - \sum_j 2^{(-\text{depth}_j/(\#\text{trees} \ \cdot \ average\_path\_length_j))}$$

$$\text{Anomaly Score}(x_i) = \begin{cases} \text{True}, & \text{if } S(x_i) \geq 0 \\ \text{False}, & \text{otherwise} \end{cases}$$

where $S(x_i)$ is the score of the i'th element. The 0.5 is the default offset as presented by the authors who created Isolation Forest (Liu, Ting, and Zhou 2008). The variable j indexes the tree where depth_{ij} is the number of ancestors of x_i in tree j. $\#trees$ is the total number of trees, and average_path_length is the avererge path length to get to the leaf for the j'th tree. Finally the anomaly score of an element is determined by if the score is greater than or equal to 0.

- Local outlier factor (LOF) (M. M. Breunig et al. 2000), computes a ratio of the local density between every element and the local density of its neighbors. The algorithm works as follows. We shall consider a point A in the set and the points B are elements of the k-Nearest Neighbors of A. We first compute local reachability density of a point A by taking the sum of the max of (distance of A to B and the distance of B to its kth nearest neighbor) and finally dividing the resulting sum by k. Given the the local reachability density (lrd) of A we compute the lrd of $A's$ k neighbors B and take the ratio of lrd(B)/(lrd(A) * k) to give us the LOF. k is a hyper-parameter that needs to be set for up to which nearest neighbor to consider. Here we set the number of neighbors considered to be 20 as the default from Pedregosa et al. 2011.

  Similar to IForest we computed this method for both logits and maximum logit and reported

the best result of the two, which turned out to be maximum logit. Finally a value of ≤ 1 is considered and inlier while a value of > 1 is considered an outlier.

- Typicality Score, computes how similar the set of output probabilities over all classes are to the average posterior distribution for a given set of classes. To construct the typicality matrix we set a threshold t and whenever a class probability exceeds t we add the probability distribution to the typicality matrix corresponding to that class 'c'. In other words, if the posterior probability for label 'c' is greater than 50%, we add the entire probability distribution to entry 'c'. We repeat this process for every image in a validation set and finally normalize each row of the typicality matrix. To test the typicality or get an anomaly score we apply a similar approach for each test image. If class 'c' of a test image exceeds threshold t we compute the distance of the current output to the typical output of class 'c'. We repeat this process for each class in the output that exceeds the threshold and take the sum of the outputs to get our anomalous score. We experimented with the thresholds as the t used for construction can be different from t used for evaluation but found it to only vary the results slightly giving extra added complexity but little benefit. Note that this method does not require labels only the knowledge that a set of examples are in-distribution.

  We interpret the typicality matrix as a course measure of what is the probability to see other classes given the presence (or belief) of class 'c'. It is possible to construct the matrix from actual data labels as opposed to the model's output, however we found that produce inferior results compared to using the output class probabilities. The final resulting matrix is of dimensions c by c. A row corresponds to the normalized sum of the output probabilities of all images of class 'c' that the model outputted a belief that class is present.

$$\text{row}_i = \begin{cases} \sum p / \sum_j p_j, & \forall p_i \geq 0.5 \\ \frac{1}{n}, & \text{if } \forall p_i < 0.5 \end{cases}$$

  Mathematically the matrix is constructed as follows. p corresponds to a concentation of

all of the probabilities per class. The i'th row is the sum of all concatenated probabilities where the i'th class has greater than 50% probability. After the summation the resulting row is normalized. If there are no such instances where that is true then the row defaults to a uniform probability distribution over all classes.

- Non-typicality Score computes an addionational matrix for dissimilarity. This score builds of off the previous Typicality Score, with an added non-typicality matrix that defines what the distribution of what objects looks like given the absence of label 'c'. This can be used in conjunction with the typicality matrix to add or subtract to the previous values. However, the addition of this matrix yielded slightly worse results, so we removed it for the final version.

Data Collection. Similar to SVSF in section 5.3.2, we fix one value for each of the four metadata attributes in the training distribution. Specifically, the DFR training set contains images with medium scale, medium occlusion, side/back viewpoint, and no zoom-in. After sampling an IID test set, we construct eight OOD test distributions by altering one attribute at a time, obtaining test sets with minimal and heavy occlusion; small and large scale; frontal and not-worn viewpoints; and medium and large zoom-in. Including the in-distribution test set, this gives us a total of nine test sets. See Appendix A.6 for details on test set sizes. Since DeepFashion Remixed is a multi-label classification task, we use sigmoid outputs. To measure performance, we calculate mAP (mean Average Precision) as a frequency-weighted average over all 13 class AP scores.

6.4 Results

Results are shown in table 6.1. We observe that the MaxLogit method outperforms the average logit and LOF by a significant margin. The MaxLogit method bears similarity to the MSP baseline (Hendrycks and Gimpel 2017), but is naturally applicable to multi-label problems. Indeed, forcing a softmax output on the multi-label logits in order to use MSP detector results in a 19.6% drop in AUROC on MS-COCO. These results establish the MaxLogit as an effective and natural baseline

	FPR95 ↓				
$\mathcal{D}_{\text{in}}$	IForest	LogitAvg	LOF	MaxLogit	Typical
VOC	98.6	98.2	84.0	35.6	**28.1**
COCO	95.6	94.5	78.4	40.4	**39.7**
	AUROC ↑				
$\mathcal{D}_{\text{in}}$	IForest	LogitAvg	LOF	MaxLogit	Typical
VOC	46.3	47.9	68.4	**90.9**	88.1
COCO	41.4	55.5	70.2	**90.3**	88.7

Table 6.1: Multi-label out-of-distribution detection comparison of the maximum logit, typicality matrix, logit average, Local Outlier Factor, and Isolation Forest anomaly detectors on PASCAL VOC and MS-COCO. The same network architecture is used for all three detectors. All results shown are percentages.

for large-scale multi-label problems. Further, the evaluation setup enables future work in out-of-distribution detection with multi-label datasets.

			Size		Occlusion		Viewpoint		Zoom	
Network	IID	OOD	Small	Large	Slight/None	Heavy	No Wear	Side/Back	Medium	Large
ResNet-50	77.6	55.1	39.4	73.0	51.5	41.2	50.5	63.2	48.7	73.3
+ ImageNet-21K *Pretraining*	80.8	58.3	40.0	73.6	55.2	43.0	63.0	67.3	50.5	73.9
+ SE (*Self-Attention*)	77.4	55.3	38.9	72.7	52.1	40.9	52.9	64.2	47.8	72.8
+ Random Erasure	78.9	56.4	39.9	75.0	52.5	42.6	53.4	66.0	48.8	73.4
+ Speckle Noise	78.9	55.8	38.4	74.0	52.6	40.8	55.7	63.8	47.8	73.6
+ Style Transfer	80.2	57.1	37.6	76.5	54.6	43.2	58.4	65.1	49.2	72.5
+ DeepAugment	79.7	56.3	38.3	74.5	52.6	42.8	54.6	65.5	49.5	72.7
+ AugMix	80.4	57.3	39.4	74.8	55.3	42.8	57.3	66.6	49.0	73.1
ResNet-152 (*Larger Models*)	80.0	57.1	40.0	75.6	52.3	42.0	57.7	65.6	48.9	74.4

Table 6.2: DeepFashion Remixed results. Unlike the previous tables, higher is better since all values are mAP scores for this multi-label classification benchmark. The "OOD" column is the average of the row's rightmost eight OOD values. All techniques do little to close the IID/OOD generalization gap.

DeepFashion Remixed. Table 6.2 shows our experimental findings on DFR, in which all evaluated methods have an average OOD mAP that is close to the baseline. In fact, most OOD mAP increases track IID mAP increases. In general, DFR's size and occlusion shifts hurt performance the most. We also evaluate with Random Erasure augmentation, which deletes rectangles within the image, to simulate occlusion (Zhong et al. 2017). Random Erasure improved occlusion performance, but Style Transfer helped even more. Nothing substantially improved OOD performance

		Scale		Occlusion		Viewpoint		Zoom	
Network	IID	Small	Large	Slight/None	Heavy	No Wear	Side/Back	Medium	Large
ResNet-101	78.7	38.0	75.2	53.7	42.9	60.7	65.4	49.8	74.0
+ ImageNet-21K *Pretraining*	80.7	38.3	74.2	51.5	43.3	59.2	68.5	50.6	73.0
+ CBAM (*Self-Attention*)	80.9	41.0	75.8	53.0	43.9	67.3	66.3	50.7	74.9
+ Random Erasure	80.1	37.4	77.6	54.8	43.7	64.9	67.5	50.4	75.4
+ Speckle Noise	79.8	38.0	73.5	51.1	43.0	63.2	65.0	49.9	73.9
+ Style Transfer	81.7	39.2	75.4	54.9	43.5	64.5	66.4	51.5	73.8
+ DeepAugment	81.3	38.4	74.7	53.3	43.3	63.2	65.9	51.0	75.3
+ AugMix	81.8	40.2	74.5	52.5	42.4	65.0	65.9	51.1	74.3
ResNet-152 (*Larger Models*)	81.0	39.7	73.5	51.2	44.2	65.1	66.1	50.3	74.1

Table 6.3: For these results we took a different partitioning of the DeepFashion Remixed dataset namely using all of the training data (excluding the same combinations we test against), and then only splitting up the validation data. All values are mAP scores. ResNet-152 tests the *Larger Models* hypothesis, ImageNet-21K Pretraining tests *Pretraining*, CBAM tests *Self-Attention*, and the other techniques test *Diverse Data Augmentation*. All techniques have limited effects.

beyond what is explained by IID performance, so here it would appear that *Only IID Accuracy Matters*. Our results do not provide clear evidence for the *Larger Models*, *Self-Attention*, *Diverse Data Augmentation*, and *Pretraining* hypotheses as discussed in chpater 5.

6.5 Conclusion

We have evaluated several classic techniques and introduced two new techniques for measuring the out-of-distribution robustness in the multi-label setting. We show that both are able achieve comparable results and extend the previous baseline of maximum softmax probability to better handle diverse and complex images.

CHAPTER 7

ROBUSTNESS IN FEW-SHOT LEARNING

7.1 Overview

Few-shot learning has gained recent popularity with the advent of novel meta-learning techniques (Finn, Abbeel, and Levine 2017). Few-shot learning is the problem of training a classification model on a task given a small number of training examples (the so-called "shots"). Due to the difficulty in generalizing from only a few instances, and to be robust to overfitting, a successful few-shot learning model must efficiently re-use what it has learned.

Given the goal of efficient reusability of learned material, we can see a clear connection between few-shot learning, and the aims of robustness. One of the aims in robustness research is to detect out-of-distribution (OOD) examples which in many cases the number of OOD examples greatly outnumbers that of in-distribution examples. In this way, the problem of few-shot learning is simply a scaling down of the original problem of OOD detection.

Previous formulations of robustness focused exclusively within the data-rich setting (Krešo et al. 2018; Tsipras et al. 2018; Orhan 2019). However, in practice there are many naturally occurring scenarios which only have a few instances. There many only exist a few instances present in either the training set, the anomalies, or even both. An example of such a phenomenon is for detecting rare species of animals (Weinstein 2018) or rare cosmic events (Ackermann et al. 2018), whereby both events are rare and collecting more data may be cost prohibitive or physically impossible.

We attempt to overcome these issues by reformulating the problem with meta-learning. Specifically we look at previous meta-learning approaches such as Simple-Shot and ProtoNet, to compare to our own approach of set-membership. We hypothesize that reformulating the problem as a set-membership task would perform better than enforcing arbitrary classification during updates to the base learner.

Meta-learning still presents challenges as the interactions between the meta-algorithm and base-learners are opaque. The extra hyperparameters and in some cases two optimizations create

an extra hurdle to determine the optimal set of parameters.

Through our experimentation, we fail to show how the reformulation of the task to set-membership benefits the problem. We hypothesize that the task may be ill-suited to learning from few examples. We experiment with different architectures, and optimization methods to arrive at our conclusion.

7.2 Set-Membership

Set-membership has a rich history (Kosut, Lau, and Boyd 1992; Gollamudi et al. 1998; Werner and Diniz 2001) and has broad applications in network protocol analysis, routing-table lookup, online traffic measurement, peer-to-peer systems, cooperative caching, firewall design, intrusion detection, bioinformatics, database QUERY processing, stream computing, and distributed storage systems (Broder and Mitzenmacher 2003).

Set-membership is the task of determining if a query element is a member of some set. Given the simple definition there are surprisingly few methods for the task. Given sufficient memory (and time) then the problem is solvable with hash-table complete dictionaries. Once memory is a constraint then approximations to the full solution need to be considered. The methods for approximation are hash compacted hash-table dictionaries, Bloom filters, and derivatives of Bloom filters.

7.3 Methods

Datasets. The miniImageNet dataset (Vinyals et al. 2016) is a subset of the popularly used ImageNet (Russakovsky et al. 2014) dataset. The dataset contains 100 classes and has a total of 600 examples per class. We follow (Ravi and Larochelle 2017) and the subsequent work to split the dataset into 64 base classes, 16 validation classes, and 20 novel classes. We pre-process the dataset as the original authors (Vinyals et al. 2016) and subsequent studies do, by resizing the images to 84×84 pixels via rescaling and center cropping.

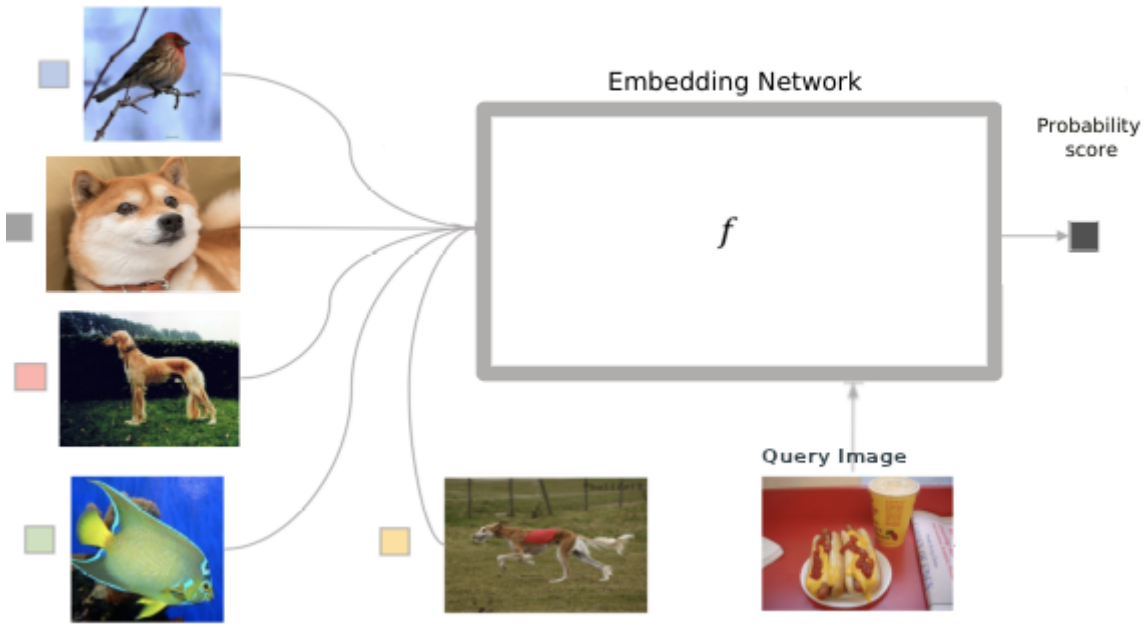


Figure 7.1: Set membership model overview. We concatenate the shot images with the single query images to feed into the network. We use different architectures as our embedding network to learn whether the query image belongs in the set of images. This is an example of a 5 shot (k=5) set membership task in which we ask if the hot dog belongs to the set of images of animals.

Evaluation protocol. To evaluate our model and the others we compared to, we conduct 10,000 classification runs of K-shot C-way tasks from the novel classes. Each task consists of selecting C of the novel classes, uses K labeled images, and 15 test images per class. For our experiments we set $K = \{1, 5\}$ for one-shot and five-shot experiments as per Vinyals et al. 2016; Yan Wang et al. 2019. For the final accuracies we average over all of the tasks and test images to report the resulting average accuracy and 95% confidence interval.

Model and Implementation details. We evaluate the following models for the set-membership problem. After we describe the neural network architectures we will describe the modifications we made to each to accomodate the new task. We study the following five network architectures following (Yan Wang et al. 2019):

Conv-4: A four-layer convolutional neural network. We follow Vinyals et al. 2016 in their implementation. The implementation consists of 4 convolution blocks of 3 x 3 filter, followed by batch normalization, rectified linear unit, and max pooling. The final layer is a linear projection to our prediction used for set-membership.

MobileNet (Howard et al. 2017): We use the same architecture as published, but we remove the first two down-sampling layers from the network.

ResNet-10/18 (He et al. 2015): We use the standard 18-layer architecture but we remove the first two down-sampling layers and we change the first convolutional layer to use a kernel of size 3 × 3 (rather than 7 × 7) pixels. Our ResNet-10 contains 4 residual blocks; the ResNet-18 contains 8 blocks.

WRN-28-10 Wide residual networks (Zagoruyko and Komodakis 2016): We use the architecture with 28 convolutional layers and a widening factor of 10.

DenseNet-121 (Huang, Liu, et al. 2017): We use the standard 121-layer architecture but similarly to MobileNet we remove the first two down-sampling layers.

Vision Transformer (Dosovitskiy et al. 2021): We use the recent Vision Transformer with the following parameters: embedding dimensions of size 1024, 6 layers deep, 8 attention heads, and a logit layer dimensions of 2048 before the final linear layer to predict the binary class. The model

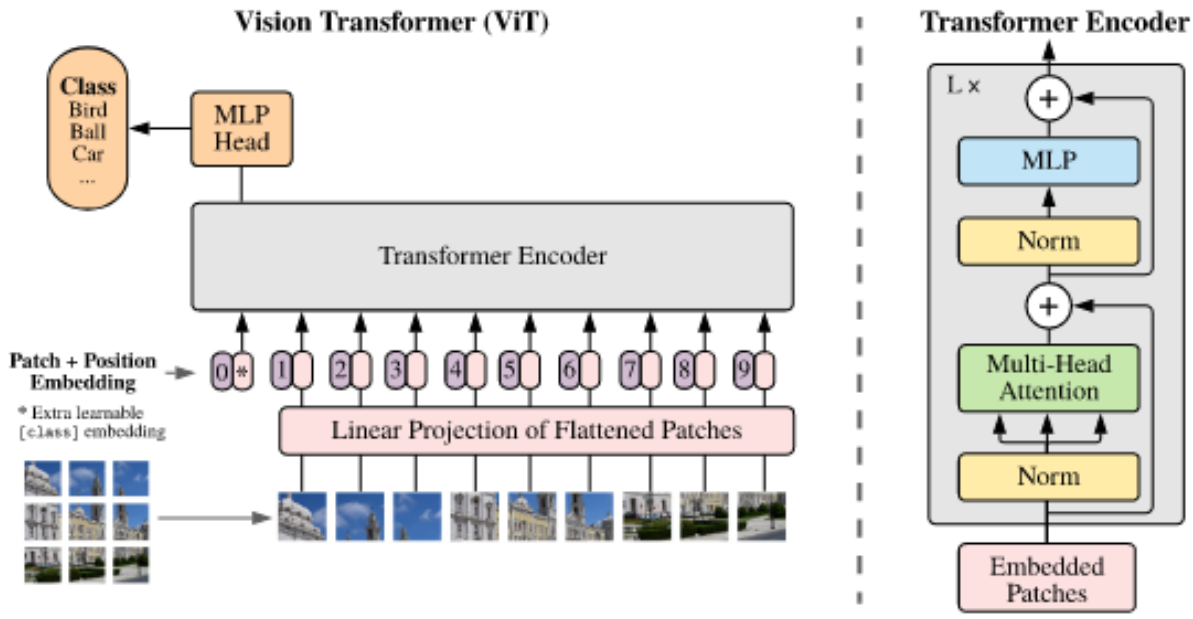


Figure 7.2: Vision Transformer model. The model takes as input non-overlapping patches of the original image whereby each are linearly embedded and position embeddings are added onto each. Finally the embeddings are fed into the Transformer along with a "classification token" to use for classification. The above reflects the original ViT. We modified the architecture from instead of accepting only 1 image it accepts a batch of images as a single query. The change corresponds to roughly increasing the model size by 6 to account for the extra 4 input images and the 1 query image. The MLP head changes to output a single number instead of n class probabilities. Image source: Dosovitskiy et al. 2021.

is depicted in figure 7.2.

Model training details. We trained all of the networks for 90 epochs from scratch using stochastic gradient descent (SGD). For all of the above networks we modified their final output layer to be a single logit which we treated as the set-membership score. We use SGD to minimize binary cross entropy of the set-membership score with the labels $\{0, 1\}$ representing negative and positive membership respectively. For SGD, we set the initial learning rate to 0.1 and decrease the learning rate by 10 at epochs 45 and 66 respectively. We use a batch size of 200 images for all of our experiments. We follow the data augmentation from He et al. 2015, which is resize, scale, shift, and horizontal flip. Another modification we made to all of the above networks is modifying the input size by concatenating the shot and query images. This would create a channel dimension of 6 and 18 for 1-shot and 5-shot respectively. Finally we employed early stopping to select the best model.

Experimental Setup. For these experiments we utilize miniImageNet. The training procedure is as follows: we sample a set of k training examples these will serve as our key set (we use $k = 1, 5$ for our experiments). Given this set of k examples from one class, sample a random image from the training set, this will serve as our query example. The label is determined by whether the example belongs to the same class or different class. During training any potential set of k examples from a given class can serve as key set.

Once the key and query sets are selected from the training data, the resulting images are concatenated together along the channel dimension. We modify the first layer of each network to accept a $(k + 1) \cdot 3$ channel image while maintaining the rest of the architecture constant. The last layer of the networks are also modified to output a single probability of inclusion or exclusion. We train for 600 iterations before retrieving a new set of keys and repeat the training procedures.

At test time we are only allowed k labeled examples per class as is commonly known as k-shot classification. There are no common classes between the test time classes and those from training. The k samples are randomly chosen per class and all of the remaining examples serve as our query examples. Finally we test all of the query examples to determine if they belong to key set. We

repeat the above procedure 10 times of choosing k labeled examples and running every other query image against our network and use the average of the results as our reported result. We set $k = 5$ for 5-shot and $k = 1$ for 1-shot classification.

We studied what effects does pretraining a model have in this setting. Specifically we used MobileNet and ResNet-18 pretrained on Places365. The reasoning for choosing this dataset to pretrain on is that it does not have any of the same classes as ImageNet so there are no conflicts with respect to the classes.

7.4 Results

Accuracy	Mini ImageNet (%) (both k=1 & k=5)
Conv-4	50
MobileNet	50
ResNet-10	50
ResNet-18	50
WRN-28-10	50
DenseNet-121	50
Vision Transformer	50
Random Baseline	50

Table 7.1: Accuracy of different architectures on miniImageNet (Vinyals et al. 2016). These results highlight that no architecture can achieve results than a random baseline.

The results suggest that the approach implemented here utilizing neural networks for the task of set-membership needs improvement as it fails to generalize to new tasks. Even with varying network architecture and optimization hyper-parameters such as SGD, Adam, weight decay and going from 1 to 5 shot seems to make no difference in generalizing from the training examples to test examples. Furthermore even utilizing a different architecture, the transformer, results in a similar poor performance which leads to conclusion that the current approach is insensitive to architecture.

The results in table 7.2 highlight how pretraining the models can improve the performance

Accuracy	Mini ImageNet (%) k=1	k=5
MobileNet	51	53
ResNet-18	53	57
Random Baseline	50	50

Table 7.2: Accuracy of different architectures on miniImageNet (Vinyals et al. 2016). These results that pretraining has some small effects on performance.

in this regime. However the results are far below that of what the previous benchmarks and performance can achieve with prototypical networks (Snell, Swersky, and Zemel 2017) for example. This highlights how pretraining helps but this has also been demonstrated by prior work.

7.5 Conclusion

We introduced a reformulation of a recent task of meta-learning into an older problem of set-membership. However, it seems that this naive reinterpretation is unsuccessful in improving in the task. We highlight some approaches and leave it as an open problem or as a warning to avoid the same pitfalls. Our analysis shows that the reinterpretation of the task as a set-membership task is insensitive to model size, and optimization scheme.

CHAPTER 8

CONCLUSIONS

We have tested and presented several novel robustness techniques throughout the paper. We began an exploration of few-shot robustness and demonstrated a how the task of set-membership as posed is ill-suited for robustness. We also have demonstrated a fragility in the loose definition of what robustness is. How in natural settings robustness is a multivariate concept that is not captured by a single metric even though the different concepts can be captured by a single word.

With the introduction of several techniques that scale better to larger images such as Maximum Logit, and Typicality Scores, we have helped the research field. Deep Augment also presents a novel augmentation technique that creates an entirely new under explored area of neural based augmentations. It remains difficult to still classify the types of augmentations learned and applied as they are data dependent augmentations.

Finally we presented several new datasets to test out our methods and techniques. Naturally Filtered Examples highlight the fragility of current models including that of transformers which is a completely different architecture type. CAOS provides a synthetic test bed for reproducible anomaly detection. Last but not least is Imagenet-R which is a representation version of a subset of ImageNet classes. Together these contributions of datasets, and techniques will hopefully advance the area of robustness research in machine learning.

APPENDIX A

A.1 ImageNet-A Classes

The 200 ImageNet classes that we selected for ImageNet-A are as follows. goldfish, great white shark, hammerhead, stingray, hen, ostrich, goldfinch, junco, bald eagle, vulture, newt, axolotl, tree frog, iguana, African chameleon, cobra, scorpion, tarantula, centipede, peacock, lorikeet, hummingbird, toucan, duck, goose, black swan, koala, jellyfish, snail, lobster, hermit crab, flamingo, american egret, pelican, king penguin, grey whale, killer whale, sea lion, chihuahua, shih tzu, afghan hound, basset hound, beagle, bloodhound, italian greyhound, whippet, weimaraner, yorkshire terrier, boston terrier, scottish terrier, west highland white terrier, golden retriever, labrador retriever, cocker spaniels, collie, border collie, rottweiler, german shepherd dog, boxer, french bulldog, saint bernard, husky, dalmatian, pug, pomeranian, chow chow, pembroke welsh corgi, toy poodle, standard poodle, timber wolf, hyena, red fox, tabby cat, leopard, snow leopard, lion, tiger, cheetah, polar bear, meerkat, ladybug, fly, bee, ant, grasshopper, cockroach, mantis, dragonfly, monarch butterfly, starfish, wood rabbit, porcupine, fox squirrel, beaver, guinea pig, zebra, pig, hippopotamus, bison, gazelle, llama, skunk, badger, orangutan, gorilla, chimpanzee, gibbon, baboon, panda, eel, clown fish, puffer fish, accordion, ambulance, assault rifle, backpack, barn, wheelbarrow, basketball, bathtub, lighthouse, beer glass, binoculars, birdhouse, bow tie, broom, bucket, cauldron, candle, cannon, canoe, carousel, castle, mobile phone, cowboy hat, electric guitar, fire engine, flute, gasmask, grand piano, guillotine, hammer, harmonica, harp, hatchet, jeep, joystick, lab coat, lawn mower, lipstick, mailbox, missile, mitten, parachute, pickup truck, pirate ship, revolver, rugby ball, sandal, saxophone, school bus, schooner, shield, soccer ball, space shuttle, spider web, steam locomotive, scarf, submarine, tank, tennis ball,

tractor, trombone, vase, violin, military aircraft, wine bottle, ice cream, bagel, pretzel, cheeseburger, hotdog, cabbage, broccoli, cucumber, bell pepper, mushroom, Granny Smith, strawberry, lemon, pineapple, banana, pomegranate, pizza, burrito, espresso, volcano, baseball player, scuba diver, acorn,

n01443537, n01484850, n01494475, n01498041, n01514859, n01518878, n01531178, n01534433, n01614925, n01616318, n01630670, n01632777, n01644373, n01677366, n01694178, n01748264, n01770393, n01774750, n01784675, n01806143, n01820546, n01833805, n01843383, n01847000, n01855672, n01860187, n01882714, n01910747, n01944390, n01983481, n01986214, n02007558, n02009912, n02051845, n02056570, n02066245, n02071294, n02077923, n02085620, n02086240, n02088094, n02088238, n02088364, n02088466, n02091032, n02091134, n02092339, n02094433, n02096585, n02097298, n02098286, n02099601, n02099712, n02102318, n02106030, n02106166, n02106550, n02106662, n02108089, n02108915, n02109525, n02110185, n02110341, n02110958, n02112018, n02112137, n02113023, n02113624, n02113799, n02114367, n02117135, n02119022, n02123045, n02128385, n02128757, n02129165, n02129604, n02130308, n02134084, n02138441, n02165456, n02190166, n02206856, n02219486, n02226429, n02233338, n02236044, n02268443, n02279972, n02317335, n02325366, n02346627, n02356798, n02363005, n02364673, n02391049, n02395406, n02398521, n02410509, n02423022, n02437616, n02445715, n02447366, n02480495, n02480855, n02481823, n02483362, n02486410, n02510455, n02526121, n02607072, n02655020, n02672831, n02701002, n02749479, n02769748, n02793495, n02797295, n02802426, n02808440, n02814860, n02823750, n02841315, n02843684, n02883205, n02906734, n02909870, n02939185, n02948072, n02950826, n02951358, n02966193, n02980441, n02992529, n03124170, n03272010, n03345487, n03372029, n03424325, n03452741, n03467068, n03481172, n03494278, n03495258, n03498962, n03594945, n03602883, n03630383, n03649909, n03676483, n03710193, n03773504, n03775071, n03888257, n03930630, n03947888, n04086273, n04118538, n04133789, n04141076, n04146614,

n04147183, n04192698, n04254680, n04266014, n04275548, n04310018, n04325704, n04347754, n04389033, n04409515, n04465501, n04487394, n04522168, n04536866, n04552348, n04591713, n07614500, n07693725, n07695742, n07697313, n07697537, n07714571, n07714990, n07718472, n07720875, n07734744, n07742313, n07745940, n07749582, n07753275, n07753592, n07768694, n07873807, n07880968, n07920052, n09472597, n09835506, n10565667, n12267677,

‘Stingray;’ ‘goldfinch, Carduelis carduelis;’ ‘junco, snowbird;’ ‘robin, American robin, Turdus migratorius;’ ‘jay;’ ‘bald eagle, American eagle, Haliaeetus leucocephalus;’ ‘vulture;’ ‘eft;’ ‘bullfrog, Rana catesbeiana;’ ‘box turtle, box tortoise;’ ‘common iguana, iguana, Iguana iguana;’ ‘agama;’ ‘African chameleon, Chamaeleo chamaeleon;’ ‘American alligator, Alligator mississipiensis;’ ‘garter snake, grass snake;’ ‘harvestman, daddy longlegs, Phalangium opilio;’ ‘scorpion;’ ‘tarantula;’ ‘centipede;’ ‘sulphur-crested cockatoo, Kakatoe galerita, Cacatua galerita;’ ‘lorikeet;’ ‘hummingbird;’ ‘toucan;’ ‘drake;’ ‘goose;’ ‘koala, koala bear, kangaroo bear, native bear, Phascolarctos cinereus;’ ‘jellyfish;’ ‘sea anemone, anemone;’ ‘flatworm, platyhelminth;’ ‘snail;’ ‘crayfish, crawfish, crawdad, crawdaddy;’ ‘hermit crab;’ ‘flamingo;’ ‘American egret, great white heron, Egretta albus;’ ‘oystercatcher, oyster catcher;’ ‘pelican;’ ‘sea lion;’ ‘Chihuahua;’ ‘golden retriever;’ ‘Rottweiler;’ ‘German shepherd, German shepherd dog, German police dog, alsatian;’ ‘pug, pug-dog;’ ‘red fox, Vulpes vulpes;’ ‘Persian cat;’ ‘lynx, catamount;’ ‘lion, king of beasts, Panthera leo;’ ‘American black bear, black bear, Ursus americanus, Euarctos americanus;’ ‘mongoose;’ ‘ladybug, ladybeetle, lady beetle, ladybird, ladybird beetle;’ ‘rhinoceros beetle;’ ‘weevil;’ ‘fly;’ ‘bee;’ ‘ant, emmet, pismire;’ ‘grasshopper, hopper;’ ‘walking stick, walkingstick, stick insect;’ ‘cockroach, roach;’ ‘mantis, mantid;’ ‘leafhopper;’ ‘dragonfly, darning needle, devil’s darning needle, sewing needle, snake feeder, snake doctor, mosquito hawk, skeeter hawk;’ ‘monarch, monarch butterfly, milkweed butterfly, Danaus plexippus;’ ‘cabbage butterfly;’ ‘lycaenid, lycaenid butterfly;’ ‘starfish, sea star;’ ‘wood rabbit, cottontail, cottontail rabbit;’ ‘porcupine, hedgehog;’ ‘fox squirrel, eastern fox squirrel, Sciurus niger;’ ‘marmot;’ ‘bison;’ ‘skunk, polecat, wood pussy;’ ‘armadillo;’ ‘baboon;’ ‘capuchin, ringtail, Cebus capucinus;’ ‘African elephant, Loxodonta africana;’ ‘puffer, pufferfish, blowfish,

globefish;' 'academic gown, academic robe, judge's robe;' 'accordion, piano accordion, squeeze box;' 'acoustic guitar;' 'airliner;' 'ambulance;' 'apron;' 'balance beam, beam;' 'balloon;' 'banjo;' 'barn;' 'barrow, garden cart, lawn cart, wheelbarrow;' 'basketball;' 'beacon, lighthouse, beacon light, pharos;' 'beaker;' 'bikini, two-piece;' 'bow;' 'bow tie, bow-tie, bowtie;' 'breastplate, aegis, egis;' 'broom;' 'candle, taper, wax light;' 'canoe;' 'castle;' 'cello, violoncello;' 'chain;' 'chest;' 'Christmas stocking;' 'cowboy boot;' 'cradle;' 'dial telephone, dial phone;' 'digital clock;' 'doormat, welcome mat;' 'drumstick;' 'dumbbell;' 'envelope;' 'feather boa, boa;' 'flagpole, flagstaff;' 'forklift;' 'fountain;' 'garbage truck, dustcart;' 'goblet;' 'go-kart;' 'golfcart, golf cart;' 'grand piano, grand;' 'hand blower, blow dryer, blow drier, hair dryer, hair drier;' 'iron, smoothing iron;' 'jack-o'-lantern;' 'jeep, landrover;' 'kimono;' 'lighter, light, igniter, ignitor;' 'limousine, limo;' 'manhole cover;' 'maraca;' 'marimba, xylophone;' 'mask;' 'mitten;' 'mosque;' 'nail;' 'obelisk;' 'ocarina, sweet potato;' 'organ, pipe organ;' 'parachute, chute;' 'parking meter;' 'piggy bank, penny bank;' 'pool table, billiard table, snooker table;' 'puck, hockey puck;' 'quill, quill pen;' 'racket, racquet;' 'reel;' 'revolver, six-gun, six-shooter;' 'rocking chair, rocker;' 'rugby ball;' 'saltshaker, salt shaker;' 'sandal;' 'sax, saxophone;' 'school bus;' 'schooner;' 'sewing machine;' 'shovel;' 'sleeping bag;' 'snowmobile;' 'snowplow, snowplough;' 'soap dispenser;' 'spatula;' 'spider web, spider's web;' 'steam locomotive;' 'stethoscope;' 'studio couch, day bed;' 'submarine, pigboat, sub, U-boat;' 'sundial;' 'suspension bridge;' 'syringe;' 'tank, army tank, armored combat vehicle, armoured combat vehicle;' 'teddy, teddy bear;' 'toaster;' 'torch;' 'tricycle, trike, velocipede;' 'umbrella;' 'unicycle, monocycle;' 'viaduct;' 'volleyball;' 'washer, automatic washer, washing machine;' 'water tower;' 'wine bottle;' 'wreck;' 'guacamole;' 'pretzel;' 'cheeseburger;' 'hotdog, hot dog, red hot;' 'broccoli;' 'cucumber, cuke;' 'bell pepper;' 'mushroom;' 'lemon;' 'banana;' 'custard apple;' 'pomegranate;' 'carbonara;' 'bubble;' 'cliff, drop, drop-off;' 'volcano;' 'ballplayer, baseball player;' 'rapeseed;' 'yellow lady's slipper, yellow lady-slipper, Cypripedium calceolus, Cypripedium parviflorum;' 'corn;' 'acorn.'

Their WordNet IDs are as follows.

n01498041, n01531178, n01534433, n01558993, n01580077, n01614925, n01616318,

n01631663, n01641577, n01669191, n01677366, n01687978, n01694178, n01698640,
n01735189, n01770081, n01770393, n01774750, n01784675, n01819313, n01820546,
n01833805, n01843383, n01847000, n01855672, n01882714, n01910747, n01914609,
n01924916, n01944390, n01985128, n01986214, n02007558, n02009912, n02037110,
n02051845, n02077923, n02085620, n02099601, n02106550, n02106662, n02110958,
n02119022, n02123394, n02127052, n02129165, n02133161, n02137549, n02165456,
n02174001, n02177972, n02190166, n02206856, n02219486, n02226429, n02231487,
n02233338, n02236044, n02259212, n02268443, n02279972, n02280649, n02281787,
n02317335, n02325366, n02346627, n02356798, n02361337, n02410509, n02445715,
n02454379, n02486410, n02492035, n02504458, n02655020, n02669723, n02672831,
n02676566, n02690373, n02701002, n02730930, n02777292, n02782093, n02787622,
n02793495, n02797295, n02802426, n02814860, n02815834, n02837789, n02879718,
n02883205, n02895154, n02906734, n02948072, n02951358, n02980441, n02992211,
n02999410, n03014705, n03026506, n03124043, n03125729, n03187595, n03196217,
n03223299, n03250847, n03255030, n03291819, n03325584, n03355925, n03384352,
n03388043, n03417042, n03443371, n03444034, n03445924, n03452741, n03483316,
n03584829, n03590841, n03594945, n03617480, n03666591, n03670208, n03717622,
n03720891, n03721384, n03724870, n03775071, n03788195, n03804744, n03837869,
n03840681, n03854065, n03888257, n03891332, n03935335, n03982430, n04019541,
n04033901, n04039381, n04067472, n04086273, n04099969, n04118538, n04131690,
n04133789, n04141076, n04146614, n04147183, n04179913, n04208210, n04235860,
n04252077, n04252225, n04254120, n04270147, n04275548, n04310018, n04317175,
n04344873, n04347754, n04355338, n04366367, n04376876, n04389033, n04399382,
n04442312, n04456115, n04482393, n04507155, n04509417, n04532670, n04540053,
n04554684, n04562935, n04591713, n04606251, n07583066, n07695742, n07697313,
n07697537, n07714990, n07718472, n07720875, n07734744, n07749582, n07753592,
n07760859, n07768694, n07831146, n09229709, n09246464, n09472597, n09835506,

n11879895, n12057211, n12144580, n12267677.

A.2 ImageNet-O Classes

The 200 ImageNet classes that we selected for ImageNet-O are as follows.

‘goldfish, Carassius auratus;’ ‘triceratops;’ ‘harvestman, daddy longlegs, Phalangium opilio;’ ‘centipede;’ ‘sulphur-crested cockatoo, Kakatoe galerita, Cacatua galerita;’ ‘lorikeet;’ ‘jellyfish;’ ‘brain coral;’ ‘chambered nautilus, pearly nautilus, nautilus;’ ‘dugong, Dugong dugon;’ ‘starfish, sea star;’ ‘sea urchin;’ ‘hog, pig, grunter, squealer, Sus scrofa;’ ‘armadillo;’ ‘rock beauty, Holocanthus tricolor;’ ‘puffer, pufferfish, blowfish, globefish;’ ‘abacus;’ ‘accordion, piano accordion, squeeze box;’ ‘apron;’ ‘balance beam, beam;’ ‘ballpoint, ballpoint pen, ballpen, Biro;’ ‘Band Aid;’ ‘banjo;’ ‘barbershop;’ ‘bath towel;’ ‘bearskin, busby, shako;’ ‘binoculars, field glasses, opera glasses;’ ‘bolo tie, bolo, bola tie, bola;’ ‘bottlecap;’ ‘brassiere, bra, bandeau;’ ‘broom;’ ‘buckle;’ ‘bulletproof vest;’ ‘candle, taper, wax light;’ ‘car mirror;’ ‘chainlink fence;’ ‘chain saw, chainsaw;’ ‘chime, bell, gong;’ ‘Christmas stocking;’ ‘cinema, movie theater, movie theatre, movie house, picture palace;’ ‘combination lock;’ ‘corkscrew, bottle screw;’ ‘crane;’ ‘croquet ball;’ ‘dam, dike, dyke;’ ‘digital clock;’ ‘dishrag, dishcloth;’ ‘dogsled, dog sled, dog sleigh;’ ‘doormat, welcome mat;’ ‘drilling platform, offshore rig;’ ‘electric fan, blower;’ ‘envelope;’ ‘espresso maker;’ ‘face powder;’ ‘feather boa, boa;’ ‘fireboat;’ ‘fire screen, fireguard;’ ‘flute, transverse flute;’ ‘folding chair;’ ‘fountain;’ ‘fountain pen;’ ‘frying pan, frypan, skillet;’ ‘golf ball;’ ‘greenhouse, nursery, glasshouse;’ ‘guillotine;’ ‘hamper;’ ‘hand blower, blow dryer, blow drier, hair dryer, hair drier;’ ‘harmonica, mouth organ, harp, mouth harp;’ ‘honeycomb;’ ‘hourglass;’ ‘iron, smoothing iron;’ ‘jack-o’-lantern;’ ‘jigsaw puzzle;’ ‘joystick;’ ‘lawn mower, mower;’ ‘library;’ ‘lighter, light, igniter, ignitor;’ ‘lipstick, lip rouge;’ ‘loupe, jeweler’s loupe;’ ‘magnetic compass;’ ‘manhole cover;’ ‘maraca;’ ‘marimba, xylophone;’ ‘mask;’ ‘matchstick;’ ‘maypole;’ ‘maze, labyrinth;’ ‘medicine chest, medicine cabinet;’ ‘mortar;’ ‘mosquito net;’ ‘mousetrap;’ ‘nail;’ ‘neck brace;’ ‘necklace;’ ‘nipple;’ ‘ocarina, sweet potato;’ ‘oil filter;’ ‘organ, pipe organ;’ ‘oscilloscope, scope, cathode-ray oscilloscope, CRO;’ ‘oxygen mask;’ ‘paddlewheel, paddle wheel;’ ‘panpipe, pandean pipe, syrinx;’ ‘park bench;’ ‘pencil sharpener;’ ‘Petri dish;’ ‘pick,

plectrum, plectron;' 'picket fence, paling;' 'pill bottle;' 'ping-pong ball;' 'pinwheel;' 'plate rack;' 'plunger, plumber's helper;' 'pool table, billiard table, snooker table;' 'pot, flowerpot;' 'power drill;' 'prayer rug, prayer mat;' 'prison, prison house;' 'punching bag, punch bag, punching ball, punchball;' 'quill, quill pen;' 'radiator;' 'reel;' 'remote control, remote;' 'rubber eraser, rubber, pencil eraser;' 'rule, ruler;' 'safe;' 'safety pin;' 'saltshaker, salt shaker;' 'scale, weighing machine;' 'screw;' 'screwdriver;' 'shoji;' 'shopping cart;' 'shower cap;' 'shower curtain;' 'ski;' 'sleeping bag;' 'slot, one-armed bandit;' 'snowmobile;' 'soap dispenser;' 'solar dish, solar collector, solar furnace;' 'space heater;' 'spatula;' 'spider web, spider's web;' 'stove;' 'strainer;' 'stretcher;' 'submarine, pigboat, sub, U-boat;' 'swimming trunks, bathing trunks;' 'swing;' 'switch, electric switch, electrical switch;' 'syringe;' 'tennis ball;' 'thatch, thatched roof;' 'theater curtain, theatre curtain;' 'thimble;' 'throne;' 'tile roof;' 'toaster;' 'tricycle, trike, velocipede;' 'turnstile;' 'umbrella;' 'vending machine;' 'waffle iron;' 'washer, automatic washer, washing machine;' 'water bottle;' 'water tower;' 'whistle;' 'Windsor tie;' 'wooden spoon;' 'wool, woolen, woollen;' 'crossword puzzle, crossword;' 'traffic light, traffic signal, stoplight;' 'ice lolly, lolly, lollipop, popsicle;' 'bagel, beigel;' 'pretzel;' 'hotdog, hot dog, red hot;' 'mashed potato;' 'broccoli;' 'cauliflower;' 'zucchini, courgette;' 'acorn squash;' 'cucumber, cuke;' 'bell pepper;' 'Granny Smith;' 'strawberry;' 'orange;' 'lemon;' 'pineapple, ananas;' 'banana;' 'jackfruit, jak, jack;' 'pomegranate;' 'chocolate sauce, chocolate syrup;' 'meat loaf, meatloaf;' 'pizza, pizza pie;' 'burrito;' 'bubble;' 'volcano;' 'corn;' 'acorn;' 'hen-of-the-woods, hen of the woods, Polyporus frondosus, Grifola frondosa.'

Their WordNet IDs are as follows.

n01443537, n01704323, n01770081, n01784675, n01819313, n01820546, n01910747, n01917289, n01968897, n02074367, n02317335, n02319095, n02395406, n02454379, n02606052, n02655020, n02666196, n02672831, n02730930, n02777292, n02783161, n02786058, n02787622, n02791270, n02808304, n02817516, n02841315, n02865351, n02877765, n02892767, n02906734, n02910353, n02916936, n02948072, n02965783, n03000134, n03000684, n03017168, n03026506, n03032252, n03075370, n03109150, n03126707, n03134739, n03160309, n03196217, n03207743, n03218198, n03223299,

n03240683, n03271574, n03291819, n03297495, n03314780, n03325584, n03344393, n03347037, n03372029, n03376595, n03388043, n03388183, n03400231, n03445777, n03457902, n03467068, n03482405, n03483316, n03494278, n03530642, n03544143, n03584829, n03590841, n03598930, n03602883, n03649909, n03661043, n03666591, n03676483, n03692522, n03706229, n03717622, n03720891, n03721384, n03724870, n03729826, n03733131, n03733281, n03742115, n03786901, n03788365, n03794056, n03804744, n03814639, n03814906, n03825788, n03840681, n03843555, n03854065, n03857828, n03868863, n03874293, n03884397, n03891251, n03908714, n03920288, n03929660, n03930313, n03937543, n03942813, n03944341, n03961711, n03970156, n03982430, n03991062, n03995372, n03998194, n04005630, n04023962, n04033901, n04040759, n04067472, n04074963, n04116512, n04118776, n04125021, n04127249, n04131690, n04141975, n04153751, n04154565, n04201297, n04204347, n04209133, n04209239, n04228054, n04235860, n04243546, n04252077, n04254120, n04258138, n04265275, n04270147, n04275548, n04330267, n04332243, n04336792, n04347754, n04371430, n04371774, n04372370, n04376876, n04409515, n04417672, n04418357, n04423845, n04429376, n04435653, n04442312, n04482393, n04501370, n04507155, n04525305, n04542943, n04554684, n04557648, n04562935, n04579432, n04591157, n04597913, n04599235, n06785654, n06874185, n07615774, n07693725, n07695742, n07697537, n07711569, n07714990, n07715103, n07716358, n07717410, n07718472, n07720875, n07742313, n07745940, n07747607, n07749582, n07753275, n07753592, n07754684, n07768694, n07836838, n07871810, n07873807, n07880968, n09229709, n09472597, n12144580, n12267677, n13052670.

	ImageNet-A (Acc %)	ImageNet-O (AUPR %)
AlexNet	1.77	15.44
SqueezeNet1.1	1.12	15.31
VGG16	2.63	16.58
VGG19	2.11	16.80
VGG19+BN	2.95	16.57
DenseNet121	2.16	16.11
ResNet-18	1.15	15.23
ResNet-34	1.87	16.00
ResNet-50	2.17	16.20
ResNet-101	4.72	17.20
ResNet-152	6.05	18.00
ResNet-50+Squeeze-and-Excite	6.17	17.52
ResNet-101+Squeeze-and-Excite	8.55	17.91
ResNet-152+Squeeze-and-Excite	9.35	18.65
Res2Net-50 (v1b)	14.59	19.50
Res2Net-101 (v1b)	21.84	22.69
Res2Net-152 (v1b)	22.4	23.90
ResNeXt-50 (32×4d)	4.81	17.60
ResNeXt-101 (32×4d)	5.85	19.60
ResNeXt-101 (32×8d)	10.2	20.51
DPN 68	3.53	17.78
DPN 98	9.15	21.10

Table A.1: Expanded ImageNet-A and ImageNet-O architecture results.

A.3 Expanded Results

A.3.1 Full Architecture Results

Full results with various architectures are in A.1.

A.3.2 Calibration

In this section we show ImageNet-A calibration results.

Uncertainty Metrics. The ℓ_2 *Calibration Error* is how we measure miscalibration. We would like classifiers that can reliably forecast their accuracy. Concretely, we want classifiers which give examples 60% confidence to be correct 60% of the time. We judge a classifier's miscalibration with

Figure A.1: A demonstration of color sensitivity. While the leftmost image is classified as "banana" with high confidence, the images with modified color are correctly classified. Not only would we like models to be more accurate, we would like them to be calibrated if they wrong.

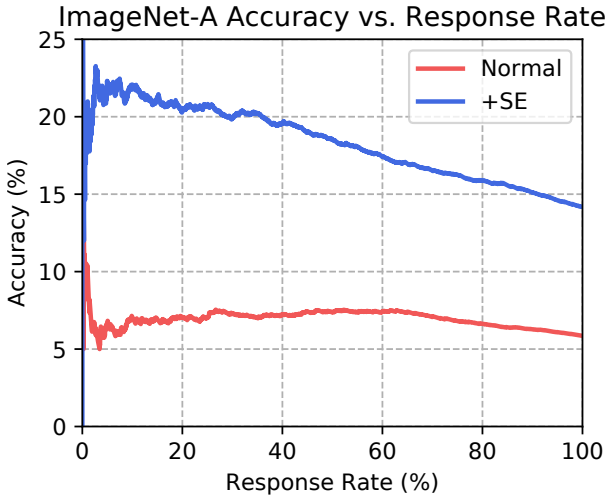


Figure A.2: The Response Rate Accuracy curve for a ResNeXt-101 (32×4d) with and without Squeeze-and-Excitation (SE). The Response Rate is the percent classified. The accuracy at a $n\%$ response rate is the accuracy on the $n\%$ of examples where the classifier is most confident.

the ℓ_2 Calibration Error Kumar, Liang, and Ma 2019.

Our second uncertainty estimation metric is the *Area Under the Response Rate Accuracy Curve (AURRA)*. Responding only when confident is often preferable to predicting falsely. In these experiments, we allow classifiers to respond to a subset of the test set and abstain from predicting the rest. Classifiers with quality uncertainty estimates should be capable identifying examples it is likely to predict falsely and abstain. If a classifier is required to abstain from predicting on 90% of the test set, or equivalently respond to the remaining 10% of the test set, then we should like the classifier's uncertainty estimates to separate correctly and falsely classified examples and have high accuracy on the selected 10%. At a fixed response rate, we should like the accuracy to be as high as possible. At a 100% response rate, the classifier accuracy is the usual test set accuracy.

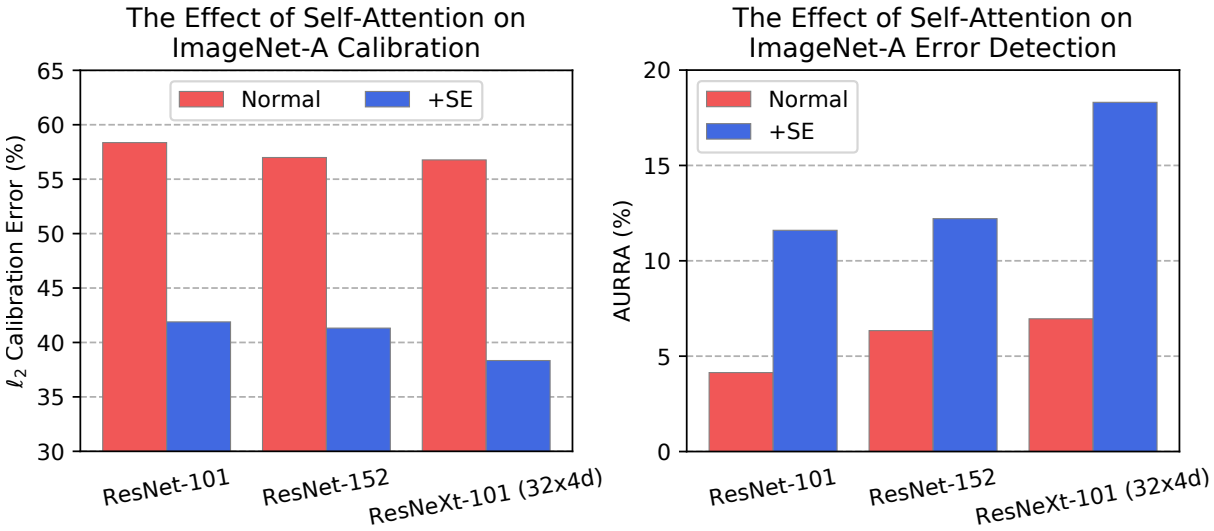


Figure A.3: Self-attention's influence on ImageNet-A ℓ_2 calibration and error detection.

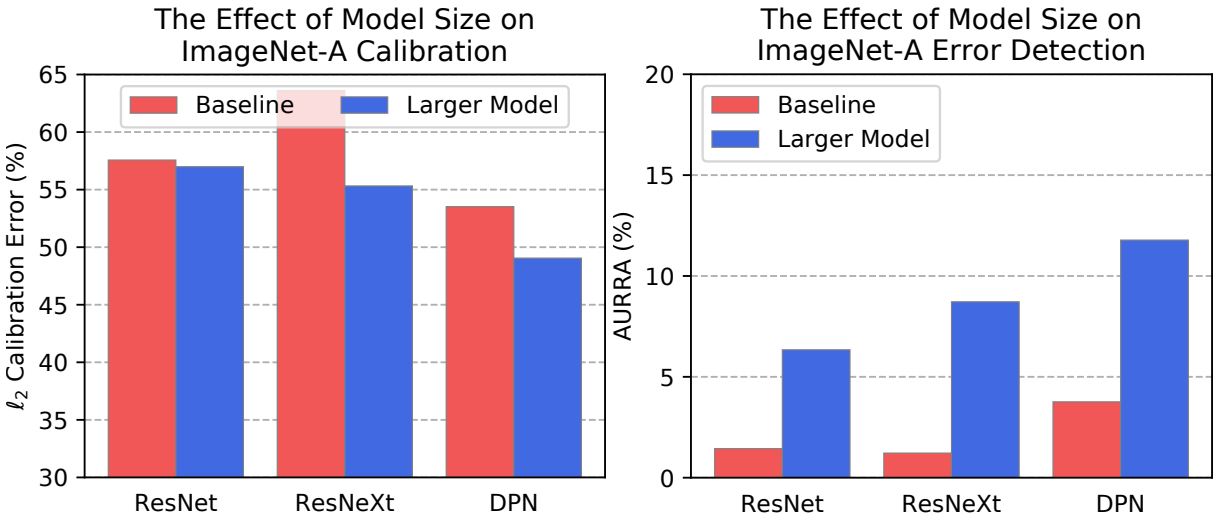


Figure A.4: Model size's influence on ImageNet-A ℓ_2 calibration and error detection.

We vary the response rates and compute the corresponding accuracies to obtain the Response Rate Accuracy (RRA) curve. The area under the Response Rate Accuracy curve is the AURRA. To compute the AURRA in this paper, we use the maximum softmax probability. For response rate p, we take the p fraction of examples with highest maximum softmax probability. If the response rate is 10%, we select the top 10% of examples with the highest confidence and compute the accuracy on these examples. An example RRA curve is in A.2 .

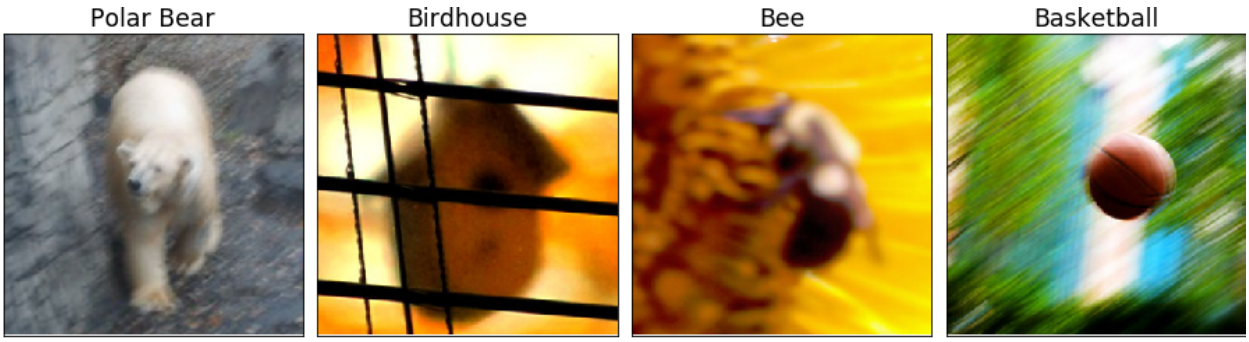


Figure A.5: Examples of real-world blurry images from our collected dataset.

A.4 Real Blurry Images and ImageNet-C

We collect 1,000 blurry images to see whether improvements on ImageNet-C's simulated blurs correspond to improvements on real-world blurry images. Each image belongs to an ImageNet class. Examples are in A.5. Results from A.2 show that *Larger Models*, *Self-Attention*, *Diverse Data Augmentation*, *Pretraining* all help, just like ImageNet-C. Here DeepAugment+AugMix attains state-of-the-art. These results suggest ImageNet-C's simulated corruptions track real-world corruptions. In hindsight, this is expected since various computer vision problems have used synthetic corruptions as proxies for real-world corruptions, for decades. In short, ImageNet-C is a diverse and systematic benchmark that is correlated with improvements on real-world corruptions.

A.5 Additional Results

ImageNet-R. Expanded ImageNet-R results are in A.4.

WSL pretraining on Instagram images appears to yield dramatic improvements on ImageNet-R, but the authors note the prevalence of artistic renditions of object classes on the Instagram platform. While ImageNet's data collection process actively excluded renditions, we do not have reason to believe the Instagram dataset excluded renditions. On a ResNeXt-101 32×8d model, WSL pretraining improves ImageNet-R performance by a massive 37.5% from 57.5% top-1 error to 24.2%. Ultimately, without examining the training images we are unable to determine whether ImageNet-R represents an actual distribution shift to the Instagram WSL models. However, we

Network	Defocus Blur	Glass Blur	Motion Blur	Zoom Blur	ImageNet-C Blur Mean	Real Blurry Images
ResNet-50	61	73	61	64	65	58.7
+ ImageNet-21K *Pretraining*	56	69	53	59	59	54.8
+ CBAM (*Self-Attention*)	60	69	56	61	62	56.5
+ ℓ_∞ Adversarial Training	80	71	72	71	74	71.6
+ Speckle Noise	57	68	60	64	62	56.9
+ Style Transfer	57	68	55	64	61	56.7
+ AugMix	52	65	46	51	54	54.4
+ DeepAugment	48	60	51	61	55	54.2
+ DeepAugment+AugMix	41	53	39	48	45	51.7
ResNet-152 (*Larger Models*)	67	81	66	74	58	54.3

Table A.2: ImageNet-C vs Real Blurry Images. All values are error rates and percentages. The rank orderings of the models on Real Blurry Images are similar to the rank orderings for "ImageNet-C Blur Mean," so ImageNet-C's simulated blurs track real-world blur performance. Hence synthetic image corruptions and real-world image corruptions are not loose and separate.

also observe that with greater controls, that is with ImageNet-21K pre-training, pretraining hardly helped ImageNet-R performance, so it is not clear that more pretraining data improves ImageNet-R performance.

Increasing model size appears to automatically improve ImageNet-R performance, as shown in A.6a. A ResNet-50 (25.5M parameters) has 63.9% error, while a ResNet-152 (60M) has 58.7% error. ResNeXt-50 32×4d (25.0M) attains 62.3% error and ResNeXt-101 32×8d (88M) attains 57.5% error.

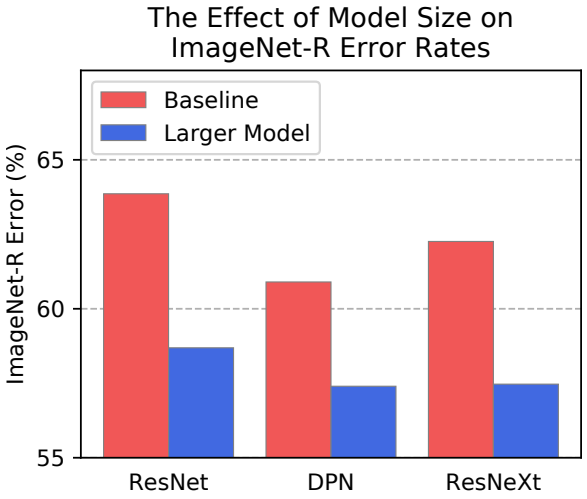
ImageNet-C. Expanded ImageNet-C results are A.5. We also tested whether model size improves performance on ImageNet-C for even larger models. With a different codebase, we trained ResNet-50, ResNet-152, and ResNet-500 models which achieved 80.6, 74.0, and 68.5 mCE respectively.

ImageNet-A. ImageNet-A Hendrycks et al. 2019 is an adversarially filtered test set. This dataset contains examples that are difficult for a ResNet-50 to classify, so examples solvable by simple spurious cues are are especially infrequent in this dataset. Results are in A.6. Notice Res2Net architectures Gao et al. 2019 can greatly improve accuracy. Results also show that *Larger Models*, *Self-Attention*, and *Pretraining* help, while *Diverse Data Augmentation* usually does not help substantially.

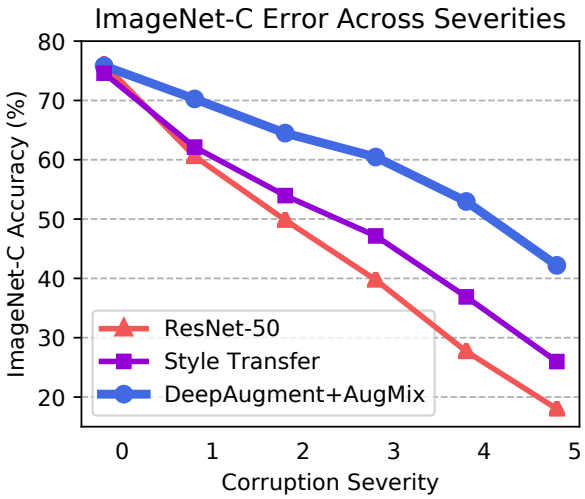
Implications for the Four Method Hypotheses.

Hypothesis	ImageNet-C	ImageNet-A	ImageNet-R	DFR	SVSF
Larger Models	+	+	+	−	
Self-Attention	+	+	−	−	
Diverse Data Augmentation	+	−	+	−	−
Pretraining	+	+	−	−	

Table A.3: A highly simplified account of each hypothesis when tested against different datasets. This table includes ImageNet-A results.



(a) Larger models improve robustness on ImageNet-R. The baseline models are ResNet-50, DPN-68, and ResNeXt-50 (32×4d). The larger models are ResNet-152, DPN-98, and ResNeXt-101 (32 × 8d). The baseline ResNeXt has a 7.1% ImageNet error rate, while the large has a 6.2% error rate.



(b) Accuracy as a function of corruption severity. Severity "0" denotes clean data. DeepAugment with AugMix shifts the entire Pareto frontier outward.

The *Larger Models* hypothesis has support with ImageNet-C (+), ImageNet-A (+), ImageNet-R (+), yet does not markedly improve DFR (−) performance.

The *Self-Attention* hypothesis has support with ImageNet-C (+), ImageNet-A (+), yet does not help ImageNet-R (−) and DFR (−) performance.

The *Diverse Data Augmentation* hypothesis has support with ImageNet-C (+), ImageNet-R (+), yet does not markedly improve ImageNet-A (−), DFR(−), nor SVSF (−) performance.

The *Pretraining* hypothesis has support with ImageNet-C (+), ImageNet-A (+), yet does not markedly improve DFR (−) nor ImageNet-R (−) performance.

	ImageNet-200 (%)	ImageNet-R (%)	Gap
ResNet-50 He et al. 2015	7.9	63.9	56.0
+ ImageNet-21K *Pretraining* (10× data)	7.0	62.8	55.8
+ CBAM (*Self-Attention*)	7.0	63.2	56.2
+ ℓ_∞ Adversarial Training	25.1	68.6	43.5
+ Speckle Noise	8.1	62.1	54.0
+ Style Transfer	8.9	58.5	49.6
+ AugMix	7.1	58.9	51.8
+ DeepAugment	7.5	57.8	50.3
+ DeepAugment + AugMix	8.0	53.2	45.2
ResNet-101 (*Larger Models*)	7.1	60.7	53.6
+ SE (*Self-Attention*)	6.7	61.0	54.3
ResNet-152 (*Larger Models*)	6.8	58.7	51.9
+ SE (*Self-Attention*)	6.6	60.0	53.4
ResNeXt-101 32×4d (*Larger Models*)	6.8	58.0	51.2
+ SE (*Self-Attention*)	5.9	59.6	53.7
ResNeXt-101 32×8d (*Larger Models*)	6.2	57.5	51.3
+ WSL *Pretraining* (1000× data)	4.1	24.2	20.1
+ DeepAugment + AugMix	6.1	47.9	41.8

Table A.4: ImageNet-200 and ImageNet-Renditions error rates. ImageNet-21K and WSL Pretraining test the *Pretraining* hypothesis, and here pretraining gives mixed benefits. CBAM and SE test the *Self-Attention* hypothesis, and these *hurt* robustness. ResNet-152 and ResNeXt-101 32×8d test the *Larger Models* hypothesis, and these help. Other methods augment data, and Style Transfer, AugMix, and DeepAugment provide support for the *Diverse Data Augmentation* hypothesis.

A.6 Further Dataset Descriptions

ImageNet-R Classes. The 200 ImageNet classes and their WordNet IDs in ImageNet-R are as follows.

Goldfish, great white shark, hammerhead, stingray, hen, ostrich, goldfinch, junco, bald eagle, vulture, newt, axolotl, tree frog, iguana, African chameleon, cobra, scorpion, tarantula, centipede, peacock, lorikeet, hummingbird, toucan, duck, goose, black swan, koala, jellyfish, snail, lobster, hermit crab, flamingo, american egret, pelican, king penguin, grey whale, killer whale, sea lion, chihuahua, shih tzu, afghan hound, basset hound, beagle, bloodhound, italian greyhound, whip-

			Noise			Blur				Weather				Digital			
	Clean	*mCE*	Gauss.	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
ResNet-50	23.9	76.7	80	82	83	75	89	78	80	78	75	66	57	71	85	77	77
+ ImageNet-21K *Pretraining*	22.4	65.8	61	64	63	69	84	68	74	69	71	61	53	53	81	54	63
+ SE (*Self-Attention*)	22.4	68.2	63	66	66	71	82	67	74	74	72	64	55	71	73	60	67
+ CBAM (*Self-Attention*)	22.4	70.0	67	68	68	74	83	71	76	73	72	65	54	70	79	62	67
+ ℓ_∞ Adversarial Training	46.2	94.0	91	92	95	97	86	92	88	93	99	118	104	111	90	72	81
+ Speckle Noise	24.2	68.3	51	47	55	70	83	77	80	76	71	66	57	70	82	72	69
+ Style Transfer	25.4	69.3	66	67	68	70	82	69	80	68	71	65	58	66	78	62	70
+ AugMix	22.5	65.3	67	66	68	64	79	59	64	69	68	65	54	57	74	60	66
+ DeepAugment	23.3	60.4	49	50	47	59	73	65	76	64	60	58	51	61	76	48	67
+ DeepAugment + AugMix	24.2	53.6	46	45	44	50	64	50	61	58	57	54	52	48	71	43	61
ResNet-152 (*Larger Models*)	21.7	69.3	73	73	76	67	81	66	74	71	68	62	51	67	76	69	65
ResNeXt-101 32×8d (*Larger Models*)	20.7	66.7	68	69	71	65	79	66	71	69	66	60	50	66	74	61	64
+ WSL *Pretraining* (1000× data)	17.8	51.7	49	50	51	53	72	55	63	53	51	42	37	41	67	40	51
+ DeepAugment + AugMix	20.1	44.5	36	35	34	43	55	42	55	48	48	47	43	39	59	34	50

Table A.5: Clean Error, Corruption Error (CE), and mean CE (mCE) values for various models and training methods on ImageNet-C. The mCE value is computed by averaging across all 15 CE values. A CE value greater than 100 (e.g. adversarial training on contrast) denotes worse performance than AlexNet. DeepAugment+AugMix improves robustness by over 23 mCE.

	ImageNet-A (%)
ResNet-50	2.2
+ ImageNet-21K *Pretraining* (10× data)	11.4
+ Squeeze-and-Excitation (*Self-Attention*)	6.2
+ CBAM (*Self-Attention*)	6.9
+ ℓ_∞ Adversarial Training	1.7
+ Style Transfer	2.0
+ AugMix	3.8
+ DeepAugment	3.5
+ DeepAugment + AugMix	3.9
ResNet-152 (*Larger Models*)	6.1
ResNet-152+Squeeze-and-Excitation (*Self-Attention*)	9.4
Res2Net-50 v1b	14.6
Res2Net-152 v1b (*Larger Models*)	22.4
ResNeXt-101 (32 × 8d) (*Larger Models*)	10.2
+ WSL *Pretraining* (1000× data)	45.4
+ DeepAugment + AugMix	11.5

Table A.6: ImageNet-A top-1 accuracy.

pet, weimaraner, yorkshire terrier, boston terrier, scottish terrier, west highland white terrier, golden retriever, labrador retriever, cocker spaniels, collie, border collie, rottweiler, german shepherd dog, boxer, french bulldog, saint bernard, husky, dalmatian, pug, pomeranian, chow chow, pembroke welsh corgi, toy poodle, standard poodle, timber wolf, hyena, red fox, tabby cat, leopard, snow leopard, lion, tiger, cheetah, polar bear, meerkat, ladybug, fly, bee, ant, grasshopper, cockroach, mantis, dragonfly, monarch butterfly, starfish, wood rabbit, porcupine, fox squirrel, beaver, guinea pig, zebra, pig, hippopotamus, bison, gazelle, llama, skunk, badger, orangutan, gorilla, chimpanzee, gibbon, baboon, panda, eel, clown fish, puffer fish, accordion, ambulance, assault rifle, backpack, barn, wheelbarrow, basketball, bathtub, lighthouse, beer glass, binoculars, birdhouse, bow tie, broom, bucket, cauldron, candle, cannon, canoe, carousel, castle, mobile phone, cowboy hat, electric guitar, fire engine, flute, gasmask, grand piano, guillotine, hammer, harmonica, harp, hatchet, jeep, joystick, lab coat, lawn mower, lipstick, mailbox, missile, mitten, parachute, pickup truck, pirate ship, revolver, rugby ball, sandal, saxophone, school bus, schooner, shield, soccer ball, space shuttle, spider web, steam locomotive, scarf, submarine, tank, tennis ball, tractor, trombone, vase, violin, military aircraft, wine bottle, ice cream, bagel, pretzel, cheeseburger, hotdog, cabbage, broccoli, cucumber, bell pepper, mushroom, Granny Smith, strawberry, lemon, pineapple, banana, pomegranate, pizza, burrito, espresso, volcano, baseball player, scuba diver, acorn.

n01443537, n01484850, n01494475, n01498041, n01514859, n01518878, n01531178, n01534433, n01614925, n01616318, n01630670, n01632777, n01644373, n01677366, n01694178, n01748264, n01770393, n01774750, n01784675, n01806143, n01820546, n01833805, n01843383, n01847000, n01855672, n01860187, n01882714, n01910747, n01944390, n01983481, n01986214, n02007558, n02009912, n02051845, n02056570, n02066245, n02071294, n02077923, n02085620, n02086240, n02088094, n02088238,

n02088364, n02088466, n02091032, n02091134, n02092339, n02094433, n02096585, n02097298, n02098286, n02099601, n02099712, n02102318, n02106030, n02106166, n02106550, n02106662, n02108089, n02108915, n02109525, n02110185, n02110341, n02110958, n02112018, n02112137, n02113023, n02113624, n02113799, n02114367, n02117135, n02119022, n02123045, n02128385, n02128757, n02129165, n02129604, n02130308, n02134084, n02138441, n02165456, n02190166, n02206856, n02219486, n02226429, n02233338, n02236044, n02268443, n02279972, n02317335, n02325366, n02346627, n02356798, n02363005, n02364673, n02391049, n02395406, n02398521, n02410509, n02423022, n02437616, n02445715, n02447366, n02480495, n02480855, n02481823, n02483362, n02486410, n02510455, n02526121, n02607072, n02655020, n02672831, n02701002, n02749479, n02769748, n02793495, n02797295, n02802426, n02808440, n02814860, n02823750, n02841315, n02843684, n02883205, n02906734, n02909870, n02939185, n02948072, n02950826, n02951358, n02966193, n02980441, n02992529, n03124170, n03272010, n03345487, n03372029, n03424325, n03452741, n03467068, n03481172, n03494278, n03495258, n03498962, n03594945, n03602883, n03630383, n03649909, n03676483, n03710193, n03773504, n03775071, n03888257, n03930630, n03947888, n04086273, n04118538, n04133789, n04141076, n04146614, n04147183, n04192698, n04254680, n04266014, n04275548, n04310018, n04325704, n04347754, n04389033, n04409515, n04465501, n04487394, n04522168, n04536866, n04552348, n04591713, n07614500, n07693725, n07695742, n07697313, n07697537, n07714571, n07714990, n07718472, n07720875, n07734744, n07742313, n07745940, n07749582, n07753275, n07753592, n07768694, n07873807, n07880968, n07920052, n09472597, n09835506, n10565667, n12267677.

SVSF. The classes are

- auto shop
- bank
- car dealer
- bakery
- beauty salon
- car wash

- cell phone store
- dentist
- discount store
- dry cleaner
- furniture store
- gas station
- gym
- hardware store
- hotel
- liquor store
- pharmacy
- religious institution
- storage facility
- veterinary care.

DeepFashion Remixed. The classes are

- short sleeve top
- long sleeve top
- short sleeve outerwear
- long sleeve outerwear
- vest
- sling
- shorts
- trousers
- skirt
- short sleeve dress
- long sleep dress
- vest dress
- sling dress.

Size (small, moderate, or large) defines how much of the image the article of clothing takes up. Occlusion (slight, medium, or heavy) defines the degree to which the object is occluded from the camera. Viewpoint (front, side/back, or not worn) defines the camera position relative to the article of clothing. Zoom (no zoom, medium, or large) defines how much camera zoom was used to take the picture.

	Represented Distribution Shifts
ImageNet-Renditions	artistic renditions (cartoons, graffiti, embroidery, graphics, origami, paintings, sculptures, sketches, tattoos, toys, ...)
DeepFashion Remixed	occlusion, size, viewpoint, zoom
StreetView StoreFronts	camera, capture year, country

Table A.7: Various distribution shifts represented in our three new benchmarks. ImageNet-Renditions is a new test set for ImageNet trained models measuring robustness to various object renditions. DeepFashion Remixed and StreetView StoreFronts each contain a training set and multiple test sets capturing a variety of distribution shifts.

	Training set	Testing images
ImageNet-R	1281167	30000
DFR	48000	42640, 7440, 28160, 10360, 480, 11040, 10520, 10640
SVSF	200000	10000, 10000, 10000, 8195, 9788

Table A.8: Number of images in each training and test set. ImageNet-R training set refers to the ILSVRC 2012 training set Russakovsky et al. 2014. DeepFashion Remixed test sets are: in-distribution, occlusion - none/slight, occlusion - heavy, size - small, size - large, viewpoint - frontal, viewpoint - not-worn, zoom-in - medium, zoom-in - large. StreetView StoreFronts test sets are: in-distribution, capture year - 2018, capture year - 2017, camera system - new, country - France.

A.7 DeepAugment Details

```
def main():
    net.apply_weights(deepAugment_getNetwork())  # EDSR, CAE, ...
    for image in dataset:  # May be the ImageNet training set
        if np.random.uniform() < 0.05:  # Arbitrary refresh prob
            net.apply_weights(deepAugment_getNetwork())
        new_image = net.deepAugment_forwardPass(image)

def deepAugment_getNetwork():
    weights = load_clean_weights()
    weight_distortions = sample_weight_distortions()
    for d in weight_distortions:
        weights = apply_distortion(d, weights)
    return weights

def sample_weight_distortions():
    distortions = [
        negate_weights,
        zero_weights,
        flip_transpose_weights,
        ...
    ]

    return random_subset(distortions)

def sample_signal_distortions():
    distortions = [
        dropout,
        gelu,
        negate_signal_random_mask,
        flip_signal,
```

```
        ...
    ]

    return random_subset(distortions)


class Network():
    def apply_weights(weights):
        # Apply given weight tensors to network
        ...

    # Clean forward pass. Compare to deepAugment_forwardPass()
    def clean_forwardPass(X):
        X = network.block1(X)
        X = network.block2(X)
        ...
        X = network.blockN(X)
        return X

    # Our forward pass. Compare to clean_forwardPass()
    def deepAugment_forwardPass(X):
        # Returns a list of distortions, each of which
        # will be applied at a different layer.
        signal_distortions = sample_signal_distortions()

        X = network.block1(X)
        apply_layer_1_distortions(X, signal_distortions)
        X = network.block2(X)
        apply_layer_2_distortions(X, signal_distortions)
        ...
        apply_layer_N-1_distortions(X, signal_distortions)
        X = network.blockN(X)
        apply_layer_N_distortions(X, signal_distortions)
```

```
64
65     return X
```

Pseudocode. Above is Pythonic pseudocode for DeepAugment. The basic structure of DeepAugment is agnostic to the backbone network used, but specifics such as which layers are chosen for various transforms may vary as the backbone architecture varies. We do not need to train many different image-to-image models to get diverse distortions R. Zhang et al. 2018; Lee et al. 2020. We only use two existing models, the EDSR super-resolution model Lim et al. 2017 and the CAE image compression model Theis et al. 2017. See full code for such details.

At a high level, we process each image with an image-to-image network. The image-to-image's weights and feedforward signal pass are distorted with each pass. The distortion is made possible by, for example, negating the network's weights and applying dropout to the feedforward signal. The resulting image is distorted and saved. This process generates an augmented dataset.

REFERENCES

Achanta, Radhakrishna, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk. 2010. *Slic superpixels.* Technical report.

Ackermann, Sandro, Kevin Schawinski, Ce Zhang, Anna K Weigel, and M Dennis Turp. 2018. "Using transfer learning to detect galaxy mergers." *Monthly Notices of the Royal Astronomical Society* 479 (1): 415–425.

Agrawal, Pulkit, Ross Girshick, and Jitendra Malik. 2014. "Analyzing the Performance of Multilayer Neural Networks for Object Recognition." *ECCV.*

Ahmed, Faruk, and Aaron C. Courville. 2019. "Detecting semantic anomalies." *ArXiv* abs/1908.04388.

Altman, Naomi S. 1992. "An Introduction to Kernel and Nearest Neighbor Nonparametric Regression."

Anguelov, Dragomir, Carole Dulong, Daniel Filip, Christian Frueh, Stéphane Lafon, Richard Lyon, Abhijit Ogale, Luc Vincent, and Josh Weaver. 2010. "Google street view: Capturing the world at street level." *Computer* 43 (6): 32–38.

Angus, Matt. 2019. *Towards Pixel-Level OOD Detection for Semantic Segmentation.*

Arazo, Eric, Diego Ortego, Paul Albert, Noel E. O'Connor, and Kevin McGuinness. 2019. *Unsupervised Label Noise Modeling and Loss Correction.* arXiv: 1904.11238 [cs.CV].

Arbelaez, Pablo, Michael Maire, Charless Fowlkes, and Jitendra Malik. 2010. "Contour detection and hierarchical image segmentation." *IEEE transactions on pattern analysis and machine intelligence* 33 (5): 898–916.

Arjovsky, Martín, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. "Invariant Risk Minimization." *ArXiv* abs/1907.02893.

Arnab, A., Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, M. Larsson, A. Kirillov, Bogdan Savchynskyy, C. Rother, F. Kahl, and P. Torr. 2018. "Conditional Random Fields Meet Deep Neural Networks for Semantic Segmentation: Combining Probabilistic Graphical Models with Deep Learning for Structured Prediction." *IEEE Signal Processing Magazine* 35:37–52.

Athalye, Anish, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2017. "Synthesizing robust adversarial examples." *arXiv preprint arXiv:1707.07397.*

Baluja, Shumeet, and Ian Fischer. 2017. "Adversarial Transformation Networks: Learning to Generate Adversarial Examples." *CoRR* abs/1703.09387.

Baur, Christoph, Benedikt Wiestler, Shadi Albarqouni, and Nassir Navab. 2019. "Deep Autoencoding Models for Unsupervised Anomaly Segmentation in Brain MR Images." *Lecture Notes in Computer Science,* 161–169. ISSN: 1611-3349. https://doi.org/10.1007/978-3-030-11723-8_16.

Beede, Emma, Elizabeth Baylor, Fred Hersch, Anna Iurchenko, Lauren Wilcox, Paisan Ruamviboonsuk, and Laura M Vardoulakis. 2020. "A Human-Centered Evaluation of a Deep Learning System Deployed in Clinics for the Detection of Diabetic Retinopathy." In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems,* 1–12.

Ben-Baruch, Emanuel, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. 2020. *Asymmetric Loss For Multi-Label Classification.* arXiv: 2009.14119 [cs.CV].

Bevandić, Petra, Ivan Krešo, Marin Oršić, and Siniša Šegvić. 2018. *Discriminative out-of-distribution detection for semantic segmentation.* arXiv: 1808.07703 [cs.CV].

Beygelzimer, A., Sham M. Kakade, and J. Langford. 2006. "Cover trees for nearest neighbor." In *ICML '06.*

Bhagavatula, Chandra, Ronan Le Bras, Chaitanya Malaviya, Keisuke Sakaguchi, Ari Holtzman, Hannah Rashkin, Doug Downey, Scott Yih, and Yejin Choi. 2019. "Abductive Commonsense Reasoning." *ArXiv* abs/1908.05739.

Bhattad, Anand, Min Chong, Kaizhao Liang, Bo Li, and David Forsyth. 2019. "Big but Imperceptible Adversarial Perturbations via Semantic Manipulation." *abs/1904.06347.*

Bhattad, Anand, Min Jin Chong, Kaizhao Liang, Bo Li, and David A Forsyth. 2020. "Unrestricted Adversarial Examples via Semantic Manipulation." In *ICLR.*

Biederman, Irving, and Ginny Ju. 1988. "Surface versus edge-based determinants of visual recognition." *Cognitive psychology* 20 (1): 38–64.

Bisk, Yonatan, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. 2019. "PIQA: Reasoning about Physical Commonsense in Natural Language." *ArXiv* abs/1911.11641.

Blum, Hermann, Paul-Edouard Sarlin, Juan Nieto, Roland Siegwart, and Cesar Cadena. 2019. *The Fishyscapes Benchmark: Measuring Blind Spots in Semantic Segmentation.* arXiv: 1904.03215 [cs.CV].

Breiman, L. 2004. "Random Forests." *Machine Learning* 45:5–32.

Brendel, Wieland, and Matthias Bethge. 2018. "Approximating CNNs with Bag-of-local-Features models works surprisingly well on ImageNet." *CoRR* abs/1904.00760.

Breunig, M., H. Kriegel, R. Ng, and J. Sander. 2000. "LOF: identifying density-based local outliers." In *SIGMOD '00.*

Breunig, Markus M, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. 2000. "LOF: identifying density-based local outliers." In *ACM sigmod record,* 29:93–104. 2. ACM.

Brhanie, Bekalu Mullu. 2016. "Multi-Label Classification Methods for Image Annotation."

Broder, A., and M. Mitzenmacher. 2003. "Network Applications of Bloom Filters: A Survey." *Internet Mathematics* 1:485–509.

Brown, Tom B, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. "Adversarial patch." *arXiv preprint arXiv:1712.09665.*

Brown, Tom B., Nicholas Carlini, Chiyuan Zhang, Catherine Olsson, Paul Francis Christiano, and Ian J. Goodfellow. 2018. "Unrestricted Adversarial Examples." *CoRR* abs/1809.08352.

Cai, Zheng, Lifu Tu, and Kevin Gimpel. 2017. "Pay Attention to the Ending: Strong Neural Baselines for the ROC Story Cloze Task." In *ACL.*

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

Carlini, Nicholas, and David A. Wagner. 2017. "Towards Evaluating the Robustness of Neural Networks." *2017 IEEE Symposium on Security and Privacy (SP),* 39–57.

Carreira, Joao, and Cristian Sminchisescu. 2010. "Constrained parametric min-cuts for automatic object segmentation." In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 3241–3248. IEEE.

Chao, Wei-Lun, Han-Jia Ye, De-Chuan Zhan, Mark Campbell, and Kilian Q. Weinberger. 2020. *Revisiting Meta-Learning as Supervised Learning.* arXiv: 2002.00573 [cs.LG].

Charikar, M., J. Steinhardt, and G. Valiant. 2017. "Learning from untrusted data." *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing.*

Chen, Tianshui, Muxin Xu, Xiaolu Hui, Hefeng Wu, and Liang Lin. 2019. *Learning Semantic-Specific Graph Representation for Multi-Label Image Recognition.* arXiv: 1908.07325 [cs.CV].

Cho, Youngmin, and L. Saul. 2009. "Kernel Methods for Deep Learning." In *NIPS.*

Cimpoi, M., S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi. 2014. "Describing Textures in the Wild." In *Computer Vision and Pattern Recognition.*

Cimpoi, Mircea, Subhransu Maji, Iasonas Kokkinos, Sammy Mohamed, and Andrea Vedaldi. 2014. "Describing Textures in the Wild." *Computer Vision and Pattern Recognition.*

Cohen, Jeremy M., Elan Rosenfeld, and J. Zico Kolter. 2019. "Certified Adversarial Robustness via Randomized Smoothing." In *ICML.*

Cordts, Marius, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. 2016. "The Cityscapes Dataset for Semantic Urban Scene Understanding." In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*

Cox, D. R. 1958. "The Regression Analysis of Binary Sequences."

Cubuk, Ekin Dogus, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. 2018. "AutoAugment: Learning Augmentation Policies from Data." *CVPR.*

Deng, J., W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. 2009. "ImageNet: A Large-Scale Hierarchical Image Database." In *CVPR09.*

Deng, Jia. 2012. *Large scale visual recognition.* Technical report. PRINCETON UNIV NJ DEPT OF COMPUTER SCIENCE.

DeVries, Terrance, and Graham W Taylor. 2018. "Learning Confidence for Out-of-Distribution Detection in Neural Networks." *arXiv preprint arXiv:1802.04865.*

Devries, Terrance, and Graham W. Taylor. 2017. "Improved Regularization of Convolutional Neural Networks with Cutout." *arXiv preprint arXiv:1708.04552.*

Dodge, Samuel, and Lina Karam. 2017. "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),* 1–7. IEEE.

Dosovitskiy, Alexey, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, et al. 2021. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale.* arXiv: 2010.11929 [cs.CV].

Dosovitskiy, Alexey, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. 2017. "CARLA: An Open Urban Driving Simulator." In *Proceedings of the 1st Annual Conference on Robot Learning,* 1–16.

Dua, Dheeru, Yizhong Wang, Pradeep Dasigi, Gabriel Stanovsky, Sameer Singh, and Matt Gardner. 2019. "DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs." In *NAACL-HLT.*

Duygulu, P., K. Barnard, J. F. G. de Freitas, and D. A. Forsyth. 2002. "Object Recognition as Machine Translation: Learning a Lexicon for a Fixed Image Vocabulary." In *Computer Vision — ECCV 2002,* edited by Anders Heyden, Gunnar Sparr, Mads Nielsen, and Peter Johansen, 97–112. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-540-47979-6.

Dziedzic, Adam, John Paparrizos, Sanjay Krishnan, Aaron J. Elmore, and Michael J. Franklin. 2019. "Band-limited Training and Inference for Convolutional Neural Networks." In *ICML.*

Emmott, Andrew, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. 2015. *A Meta-Analysis of the Anomaly Detection Problem.* arXiv: 1503.01158 [cs.AI].

Emmott, Andrew F, Shubhomoy Das, Thomas Dietterich, Alan Fern, and Weng-Keen Wong. 2013. "Systematic construction of anomaly detection benchmarks from real data." In *Proceedings of the ACM SIGKDD workshop on outlier detection and description,* 16–21.

Engstrom, Logan, Justin Gilmer, Gabriel Goh, Dan Hendrycks, Andrew Ilyas, Aleksander Madry, Reiichiro Nakano, et al. 2019. "A Discussion of 'Adversarial Examples Are Not Bugs, They Are Features'." Https://distill.pub/2019/advex-bugs-discussion, *Distill,* https://doi.org/10.23915/distill.00019.

Engstrom, Logan, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Jacob Steinhardt, and Aleksander Madry. 2020. "Identifying Statistical Bias in Dataset Replication." *ICML.*

Everingham, Mark, Luc Van Gool, Christopher K. I. Williams, John M. Winn, and Andrew Zisserman. 2009. "The Pascal Visual Object Classes (VOC) Challenge." *International Journal of Computer Vision* 88:303–338.

Eykholt, Kevin, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Florian Tramèr, Atul Prakash, Tadayoshi Kohno, and Dawn Xiaodong Song. 2018. "Physical Adversarial Examples for Object Detectors." *ArXiv* abs/1807.07769.

Finn, Chelsea, Pieter Abbeel, and Sergey Levine. 2017. "Model-agnostic meta-learning for fast adaptation of deep networks." *arXiv preprint arXiv:1703.03400.*

Fischler, M., and R. Bolles. 1981. "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography." *Commun. ACM* 24:381–395.

Franceschi, Luca, Paolo Frasconi, Saverio Salzo, Riccardo Grazzi, and Massimilano Pontil. 2018. *Bilevel Programming for Hyperparameter Optimization and Meta-Learning.* arXiv: 1806.04910 [stat.ML].

Gal, Yarin, and Zoubin Ghahramani. 2016. "Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning." *International Conference on Machine Learning.*

Gao, Shanghua, Ming-Ming Cheng, Kai Zhao, Xinyu Zhang, Ming-Hsuan Yang, and Philip H. S. Torr. 2019. "Res2Net: A New Multi-scale Backbone Architecture." *IEEE transactions on pattern analysis and machine intelligence.*

Ge, Yuying, Ruimao Zhang, Xiaogang Wang, Xiaoou Tang, and Ping Luo. 2019. "Deepfashion2: A versatile benchmark for detection, pose estimation, segmentation and re-identification of clothing images." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 5337–5345.

Geirhos, Robert, Jorn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020a. "Shortcut Learning in Deep Neural Networks." *ArXiv* abs/2004.07780.

Geirhos, Robert, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. 2020b. "Shortcut Learning in Deep Neural Networks." *arXiv preprint arXiv:2004.07780.*

Geirhos, Robert, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. 2019. "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness." *ICLR.*

Geirhos, Robert, Carlos R. M. Temme, Jonas Rauber, Heiko H. Schütt, Matthias Bethge, and Felix A. Wichmann. 2018. "Generalisation in humans and deep neural networks." *NeurIPS.*

Gilmer, Justin, Ryan P. Adams, Ian J. Goodfellow, David Andersen, and George E. Dahl. 2018. "Motivating the Rules of the Game for Adversarial Example Research." *CoRR* abs/1807.06732.

Gilmer, Justin, Nicolas Ford, Nicholas Carlini, and Ekin Cubuk. 2019. "Adversarial Examples Are a Natural Consequence of Test Error in Noise," edited by Kamalika Chaudhuri and Ruslan Salakhutdinov, 97:2280–2289. Proceedings of Machine Learning Research. Long Beach, California, USA: PMLR, June. http://proceedings.mlr.press/v97/gilmer19a.html.

Gollamudi, S., S. Nagaraj, S. Kapoor, and Yih-Fang Huang. 1998. "Set-membership filtering and a set-membership normalized LMS algorithm with an adaptive step size." *IEEE Signal Processing Letters* 5:111–114.

Goodfellow, I. J., Y. Bulatov, J. Ibarz, S. Arnoud, and V. Shet. 2013. "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks." *ArXiv e-prints* (December). arXiv: 1312.6082 [cs.CV].

Goodfellow, Ian, Nicolas Papernot, Sandy Huang, Yan Duan, and Peter Abbeel. 2017. "Attacking Machine Learning with Adversarial Examples." *OpenAI Blog.*

Goodfellow, Ian J, Jonathon Shlens, and Christian Szegedy. 2014. "Explaining and harnessing adversarial examples." *arXiv preprint arXiv:1412.6572.*

Goodfellow, Ian J. 2016. *Adversarial Examples and Adversarial Training.* http://engineering.purdue.edu/~mark/puthesis.

Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. 2015. "Explaining and Harnessing Adversarial Examples." *CoRR* abs/1412.6572.

Gururangan, Suchin, Swabha Swayamdipta, Omer Levy, Roy Schwartz, Samuel R. Bowman, and Noah A. Smith. 2018. "Annotation Artifacts in Natural Language Inference Data." *ArXiv* abs/1803.02324.

Hariharan, B., P. Arbeláez, R. Girshick, and J. Malik. 2014. "Hypercolumns for Object Segmentation and Fine-grained Localization." *ArXiv e-prints* (November). arXiv: 1411.5752 [cs.CV].

Haselmann, Matthias, Dieter P Gruber, and Paul Tabatabai. 2018. "Anomaly Detection Using Deep Learning Based Image Completion." In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA),* 1237–1242. IEEE.

He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. "Mask r-cnn." In *Proceedings of the IEEE international conference on computer vision,* 2961–2969.

He, Kaiming, Xiangyu Zhang, and Shaoqing Ren. 2015. "Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification." *IEEE International Conference on Computer Vision (ICCV).*

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. "Deep Residual Learning for Image Recognition." *CoRR* abs/1512.03385. http://arxiv.org/abs/1512.03385.

Hendrycks, Dan, and Thomas Dietterich. 2019. "Benchmarking Neural Network Robustness to Common Corruptions and Perturbations." *ICLR.*

Hendrycks, Dan, and Kevin Gimpel. 2017. "A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks." *ICLR* abs/1610.02136.

Hendrycks, Dan, Kimin Lee, and Mantas Mazeika. 2019. "Using Pre-Training Can Improve Model Robustness and Uncertainty." In *ICML.*

Hendrycks, Dan, Xiaoyuan Liu, Eric Wallace, Adam Dziedzic, Rishabh Krishnan, and Dawn Song. 2020. "Pretrained Transformers Improve Out-of-Distribution Robustness." *ACL.*

Hendrycks, Dan, Mantas Mazeika, and Thomas Dietterich. 2019. "Deep Anomaly Detection with Outlier Exposure." *ICLR.*

Hendrycks, Dan, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. 2020. "AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty." *ICLR.*

Hendrycks, Dan, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. 2019. "Natural Adversarial Examples." *ArXiv* abs/1907.07174.

Hopfield, John J. 1988. "Neural networks and physical systems with emergent collective computational abilities."

Hospedales, Timothy, Antreas Antoniou, Paul Micaelli, and Amos Storkey. 2020. *Meta-Learning in Neural Networks: A Survey.* arXiv: 2004.05439 [cs.LG].

Hosseini, Hossein, and Radha Poovendran. 2018. "Semantic Adversarial Examples." *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),* 1695–16955.

Howard, Andrew G., Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications.* arXiv: 1704.04861 [cs.CV].

Hu, Jie, Li Shen, Samuel Albanie, Gang Sun, and Andrea Vedaldi. 2018. "Gather-Excite : Exploiting Feature Context in Convolutional Neural Networks." In *NeurIPS.*

Hu, Jie, Li Shen, and Gang Sun. 2018. "Squeeze-and-Excitation Networks." *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.*

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

Huang, Jonathan, Vivek Rathod, Chen Sun, Menglong Zhu, Anoop Korattikara, Alireza Fathi, Ian Fischer, Zbigniew Wojna, Yang Song, Sergio Guadarrama, et al. 2017. "Speed/accuracy trade-offs for modern convolutional object detectors." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 7310–7311.

Ilyas, Andrew, Shibani Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. 2019. "Adversarial Examples Are Not Bugs, They Are Features." In *NeurIPS.*

Itakura, Shoji. 1994. "Recognition of Line-Drawing Representations by a Chimpanzee." *The Journal of General Psychology* 121, no. 3 (July): 189–197. https://doi.org/10.1080/00221309.1994.9921195.

Kang, Daniel, Yi Sun, Dan Hendrycks, Tom Brown, and Jacob Steinhardt. 2019. "Testing Robustness Against Unforeseen Adversaries." *CoRR* abs/1908.08016.

Kendall, Alex, Vijay Badrinarayanan, and Roberto Cipolla. 2015. "Bayesian SegNet: Model Uncertainty in Deep Convolutional Encoder-Decoder Architectures for Scene Understanding." *ArXiv* abs/1511.02680.

Kingma, Diederik P., and Jimmy Ba. 2014. "Adam: A Method for Stochastic Optimization." *ICLR.*

Kirillov, Alexander, Kaiming He, Ross B. Girshick, Carsten Rother, and Piotr Dollár. 2019. "Panoptic Segmentation." *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 9396–9405.

Kirillov, Alexander, Yuxin Wu, Kaiming He, and Ross Girshick. 2019. *PointRend: Image Segmentation as Rendering.* arXiv: 1912.08193 [cs.CV].

Kolesnikov, Alexander, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. 2019. "Large Scale Learning of General Visual Representations for Transfer." *arXiv preprint arXiv:1912.11370.*

Kornblith, Simon, Jonathon Shlens, and Quoc V. Le. 2018. "Do Better ImageNet Models Transfer Better?" *CoRR* abs/1805.08974.

Kosut, R., M. Lau, and Stephen P. Boyd. 1992. "Set-membership identification of systems with parametric and nonparametric uncertainty."

Krešo, Ivan, Marin Oršić, Petra Bevandić, and Siniša Šegvić. 2018. *Robust Semantic Segmentation with Ladder-DenseNet Models.* arXiv: 1806.03465 [cs.CV].

Krizhevsky, A., and G. Hinton. 2009. "Learning multiple layers of features from tiny images." *Tech Report.*

Krizhevsky, A., Ilya Sutskever, and Geoffrey E. Hinton. 2017. "ImageNet classification with deep convolutional neural networks." In *CACM.*

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." *NIPS.*

Kumar, A., P. Liang, and T. Ma. 2019. "Verified Uncertainty Calibration." In *Advances in Neural Information Processing Systems (NeurIPS).*

Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. 2017. "Adversarial Machine Learning at Scale." *ICLR.*

Kurakin, Alexey, Ian J. Goodfellow, and Samy Bengio. 2017. "Adversarial examples in the physical world." *ArXiv* abs/1607.02533.

Kuznetsova, Alina, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, et al. 2018. "The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale." *arXiv:1811.00982.*

Lapuschkin, Sebastian, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2019. "Unmasking Clever Hans predictors and assessing what machines really learn." In *Nature Communications.*

Lecun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. "Gradient-based learning applied to document recognition." In *Proceedings of the IEEE,* 2278–2324.

Lee, Kimin, Honglak Lee, Kibok Lee, and Jinwoo Shin. 2018a. "Training Confidence-calibrated Classifiers for Detecting Out-of-Distribution Samples." *ICLR.*

Lee, Kimin, Kibok Lee, H Lee, and Jinwoo Shin. 2018b. "A Simple Unified Framework for Detecting Out-of-Distribution Samples and Adversarial Attacks." In *NeurIPS.*

Lee, Kimin, Kibok Lee, Jinwoo Shin, and Honglak Lee. 2020. "Network Randomization: A Simple Technique for Generalization in Deep Reinforcement Learning." In *ICLR.*

Li, Bo-Yi, Felix Wu, Ser-Nam Lim, Serge J. Belongie, and Kilian Q. Weinberger. 2020. "On Feature Normalization and Data Augmentation." *ArXiv* abs/2002.11102.

Liang, Shiyu, Yixuan Li, and R. Srikant. 2018. "Enhancing The Reliability of Out-of-distribution Image Detection in Neural Networks." *arXiv: Learning.*

Lim, Bee, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. 2017. "Enhanced deep residual networks for single image super-resolution." In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops,* 136–144.

Lin, Tsung-Yi, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollar. 2014. "Microsoft COCO: Common Objects in Context." *ECCV.*

Liu, F., K. Ting, and Z. Zhou. 2008. "Isolation Forest." *2008 Eighth IEEE International Conference on Data Mining*, 413–422.

Liu, Z., Qi Liu, T. Liu, Yanzhi Wang, and W. Wen. 2019. "Feature Distillation: DNN-Oriented JPEG Compression Against Adversarial Examples." *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),* 860–868.

Lopes, Raphael Gontijo, Dong Yin, Ben Poole, Justin Gilmer, and Ekin Dogus Cubuk. 2019. "Improving Robustness Without Sacrificing Accuracy with Patch Gaussian Augmentation." *arXiv preprint arXiv:1906.02611.*

Madry, A., Aleksandar Makelov, L. Schmidt, D. Tsipras, and Adrian Vladu. 2018. "Towards Deep Learning Models Resistant to Adversarial Attacks." *ArXiv* abs/1706.06083.

Madry, Aleksander, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2018. "Towards Deep Learning Models Resistant to Adversarial Attacks." *ICLR.*

Mahajan, Dhruv, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri abd Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. 2018. "Exploring the Limits of Weakly Supervised Pretraining." *ECCV.*

Martin, David, Charless Fowlkes, Doron Tal, and Jitendra Malik. 2001. "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics." In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001,* 2:416–423. IEEE.

Martin, David R, Charless C Fowlkes, and Jitendra Malik. 2004. "Learning to detect natural image boundaries using local brightness, color, and texture cues." *IEEE transactions on pattern analysis and machine intelligence* 26 (5): 530–549.

Maurer, Andreas. 2005. "Algorithmic stability and meta-learning." *Journal of Machine Learning Research* 6 (Jun): 967–994.

Meinke, Alexander, and Matthias Hein. 2019. "Towards neural networks that provably know when they don't know." *ArXiv* abs/1909.12180.

Menard, Scott. 2002. *Applied logistic regression analysis.* Vol. 106. Sage.

Mildenhall, Ben, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. 2020. "NeRF: Representing Scenes as Neural Radiance Fields for View Synthesis." In *ECCV.*

Mordvintsev, Alexander, Christopher Olah, and Mike Tyka. 2015. “Inceptionism: Going deeper into neural networks.” *arXiv.*

Mostajabi, Mohammadreza, Payman Yadollahpour, and Gregory Shakhnarovich. 2015. “Feedforward semantic segmentation with zoom-out features.” In *CVPR.*

Nam, Jinseok, Eneldo Loza Mencía, Hyunwoo J Kim, and Johannes Fürnkranz. 2017. “Maximizing Subset Accuracy with Recurrent Neural Networks in Multi-label Classification.” In *Advances in Neural Information Processing Systems 30,* edited by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, 5413–5423. Curran Associates, Inc. http://papers.nips.cc/paper/7125-maximizing-subset-accuracy-with-recurrent-neural-networks-in-multi-label-classification.pdf.

Nguyen, Anh Mai, Jason Yosinski, and Jeff Clune. 2015. “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images.” *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 427–436.

Olah, Chris, Alexander Mordvintsev, and Ludwig Schubert. 2017. “Feature Visualization: How neural networks build up their understanding of images.” Https://distill.pub/2017/feature-visualization, *Distill,* https://doi.org/10.23915/distill.00007.

Orhan, A. Emin. 2019. “Robustness properties of Facebook’s ResNeXt WSL models.” *arxiv:1907.07640.*

Patrini, Giorgio, Alessandro Rozza, Aditya Menon, Richard Nock, and Lizhen Qu. 2017. “Making Deep Neural Networks Robust to Label Noise: a Loss Correction Approach.” *CVPR,* 1944–1952.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, et al. 2011. “Scikit-learn: Machine Learning in Python.” *Journal of Machine Learning Research* 12:2825–2830.

Pinggera, Peter, Sebastian Ramos, Stefan K. Gehrig, Uwe Franke, Carsten Rother, and Rudolf Mester. 2016. “Lost and Found: Detecting Small Road Hazards for Self-Driving Vehicles.” *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS),* 1099–1106.

Ranzato, Marc’Aurelio, Christopher S. Poultney, S. Chopra, and Y. LeCun. 2006. “Efficient Learning of Sparse Representations with an Energy-Based Model.” In *NIPS.*

Ravi, S., and H. Larochelle. 2017. “Optimization as a Model for Few-Shot Learning.” In *ICLR.*

Recht, Benjamin, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. “Do ImageNet Classifiers Generalize to ImageNet?” *ArXiv* abs/1902.10811.

Ren, Mengye, Wenyuan Zeng, B. Yang, and R. Urtasun. 2018. “Learning to Reweight Examples for Robust Deep Learning.” *ArXiv* abs/1803.09050.

Rolnick, D., Andreas Veit, Serge J. Belongie, and N. Shavit. 2017. “Deep Learning is Robust to Massive Label Noise.” *ArXiv* abs/1705.10694.

Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. 2015. "U-Net: Convolutional Networks for Biomedical Image Segmentation." *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015,* 234–241. ISSN: 1611-3349. https://doi.org/10.1007/978-3-319-24574-4_28.

Rusak, Evgenia, Lukas Schott, Roland Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. 2020. "Increasing the robustness of DNNs against image corruptions by playing the Game of Noise." *arXiv preprint arXiv:2001.06057.*

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. 2014. "ImageNet Large Scale Visual Recognition Challenge." *CoRR* abs/1409.0575. http://arxiv.org/abs/1409.0575.

Saito, Takaya, and Marc Rehmsmeier. 2015. "The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets." In *PLoS ONE.*

Sakaguchi, Keisuke, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. "WINOGRANDE: An Adversarial Winograd Schema Challenge at Scale." *ArXiv* abs/1907.10641.

Savva, Manolis, Angel X. Chang, and Pat Hanrahan. 2015. "Semantically-Enriched 3D Models for Common-sense Knowledge." *CVPR 2015 Workshop on Functionality, Physics, Intentionality and Causality.*

Schölkopf, B., R. Williamson, A. Smola, John Shawe-Taylor, and John C. Platt. 1999a. "Support Vector Method for Novelty Detection." In *NIPS.*

Schölkopf, Bernhard, Robert Williamson, Alex Smola, John Shawe-Taylor, and John Platt. 1999b. "Support Vector Method for Novelty Detection." In *Proceedings of the 12th International Conference on Neural Information Processing Systems,* 582–588. NIPS'99. Denver, CO: MIT Press.

Selvaraju, Ramprasaath R., Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2019. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." *International Journal of Computer Vision* 128, no. 2 (October): 336–359. ISSN: 1573-1405. https://doi.org/10.1007/s11263-019-01228-7. http://dx.doi.org/10.1007/s11263-019-01228-7.

Selvaraju, Ramprasaath R., Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. 2019. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization." *International Journal of Computer Vision* 128:336–359.

Shapiro, Linda G., and George C. Stockman. 2001. *Computer vision.* 279–325. Prentice Hall.

Sharif, Mahmood, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. 2016. "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition." In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security,* 1528–1540. ACM.

Silla, Carlos N., and Alex Alves Freitas. 2010. "A survey of hierarchical classification across different application domains." *Data Mining and Knowledge Discovery* 22:31–72.

Snell, Jake, Kevin Swersky, and Richard S. Zemel. 2017. *Prototypical Networks for Few-shot Learning.* arXiv: 1703.05175 [cs.LG].

Song, Yang, Rui Shu, Nate Kushman, and Stefano Ermon. 2018. "Constructing Unrestricted Adversarial Examples with Generative Models." In *NeurIPS.*

Stock, Pierre, and Moustapha Cissé. 2018. "ConvNets and ImageNet Beyond Accuracy: Understanding Mistakes and Uncovering Biases." In *ECCV.*

Sun, Chen, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. 2017. "Revisiting Unreasonable Effectiveness of Data in Deep Learning Era." *ICCV.*

Sung, Kah Kay. 1995. "Learning and example selection for object and pattern detection."

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. "Intriguing properties of neural networks." *arXiv preprint arXiv:1312.6199.*

Szegedy, Christian, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2014. "Intriguing properties of neural networks." *CoRR* abs/1312.6199.

Tanaka, Masayuki. 2006. "Recognition of pictorial representations by chimpanzees (Pan troglodytes)." *Animal Cognition* 10, no. 2 (December): 169–179. https://doi.org/10.1007/s10071-006-0056-1.

Taori, Rohan, Achal Dave, V. Shankar, N. Carlini, B. Recht, and L. Schmidt. 2020a. "Measuring Robustness to Natural Distribution Shifts in Image Classification." *ArXiv* abs/2007.00644.

Taori, Rohan, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. 2020b. *When Robustness Doesn't Promote Robustness: Synthetic vs. Natural Distribution Shifts on ImageNet.* https://openreview.net/forum?id=HyxPIyrFvH.

Theis, Lucas, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. 2017. "Lossy image compression with compressive autoencoders." *arXiv preprint arXiv:1703.00395.*

Tidake, Vaishali S, and Shirish S Sane. 2018. "Multi-label Classification: a survey." *International Journal of Engineering and Technology* 7 (1045).

Tramèr, Florian, Jens Behrmann, N. Carlini, Nicolas Papernot, and Jorn-Henrik Jacobsen. 2020. "Fundamental Tradeoffs between Invariance and Sensitivity to Adversarial Perturbations." *ArXiv* abs/2002.04599.

Tramèr, Florian, N. Carlini, W. Brendel, and A. Madry. 2020. "On Adaptive Attacks to Adversarial Example Defenses." *ArXiv* abs/2002.08347.

Tramèr, Florian, A. Kurakin, Nicolas Papernot, D. Boneh, and P. McDaniel. 2018. "Ensemble Adversarial Training: Attacks and Defenses." *ArXiv* abs/1705.07204.

Tsipras, Dimitris, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. 2018. "Robustness may be at odds with accuracy." *arXiv preprint arXiv:1805.12152.*

Vinyals, Oriol, Charles Blundell, T. Lillicrap, K. Kavukcuoglu, and Daan Wierstra. 2016. "Matching Networks for One Shot Learning." *ArXiv* abs/1606.04080.

Wang, Haohan, Songwei Ge, Eric P. Xing, and Zachary C. Lipton. 2019. *Learning Robust Global Representations by Penalizing Local Predictive Power.*

Wang, Jiang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. "Cnn-rnn: A unified framework for multi-label image classification." In *Proceedings of the IEEE conference on computer vision and pattern recognition,* 2285–2294.

Wang, Ya, Dongliang He, Fu Li, Xiang Long, Zhichao Zhou, Jinwen Ma, and Shilei Wen. 2019. *Multi-Label Classification with Label Graph Superimposing.* arXiv: 1911.09243 [cs.CV].

Wang, Yan, Wei-Lun Chao, Kilian Q. Weinberger, and Laurens van der Maaten. 2019. "SimpleShot: Revisiting Nearest-Neighbor Classification for Few-Shot Learning." *arXiv preprint arXiv:1911.04623.*

Watanabe, Toshihiko. 2013. "A fuzzy RANSAC algorithm based on reinforcement learning concept." *2013 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE),* 1–6.

Weinstein, Ben G. 2018. "A computer vision for animal ecology." *Journal of Animal Ecology* 87 (3): 533–545.

Werner, S., and P.S.R. Diniz. 2001. "Set-membership affine projection algorithm." *IEEE Signal Processing Letters* 8:231–235.

Wong, Eric, and Zico Kolter. 2018. "Provable defenses against adversarial examples via the convex outer adversarial polytope." In *International Conference on Machine Learning,* 5286–5295. PMLR.

Wong, Eric, Leslie Rice, and J Zico Kolter. 2020. "Fast is better than free: Revisiting adversarial training." *arXiv preprint arXiv:2001.03994.*

Woo, Sanghyun, Jongchan Park, Joon-Young Lee, and In So Kweon. 2018. "Cbam: Convolutional block attention module." In *Proceedings of the European Conference on Computer Vision (ECCV),* 3–19.

Wu, Baoyuan, Weidong Chen, Yanbo Fan, Yong Zhang, Jinlong Hou, Junzhou Huang, Wei Liu, and Tong Zhang. 2019. "Tencent ML-Images: A Large-Scale Multi-Label Image Database for Visual Representation Learning." *arXiv preprint arXiv:1901.01703.*

Wu, Xiaoyu, and Yangsheng Wang. 2008. "Interactive foreground/background segmentation based on graph cut." In *2008 Congress on Image and Signal Processing*, 3:692–696. IEEE.

Xiao, Chaowei, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Xiaodong Song. 2018. "Spatially Transformed Adversarial Examples." *CoRR* abs/1801.02612.

Xiao, Jianxiong, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. 2010. "SUN database: Large-scale scene recognition from abbey to zoo." *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* 3485–3492.

Xie, Cihang, and Alan Yuille. 2020. "Intriguing Properties of Adversarial Training at Scale." In *International Conference on Learning Representations.*

Xie, Saining, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. 2016. "Aggregated Residual Transformations for Deep Neural Networks." *CVPR.*

Yakura, Hiromu, and Jun Sakuma. 2019. "Robust Audio Adversarial Example for a Physical Attack." *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence* (August). https://doi.org/10.24963/ijcai.2019/741. http://dx.doi.org/10.24963/ijcai.2019/741.

Yin, Dong, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D Cubuk, and Justin Gilmer. 2019. "A Fourier perspective on model robustness in computer vision." *arXiv preprint arXiv:1906.08988.*

You, Zhonghui, Jinmian Ye, Kunming Li, and Ping Wang. 2019. "Adversarial Noise Layer: Regularize Neural Network by Adding Noise." *2019 IEEE International Conference on Image Processing (ICIP),* 909–913.

Yu, Fisher, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. 2018. "BDD100K: A Diverse Driving Video Database with Scalable Annotation Tooling." *CoRR* abs/1805.04687. arXiv: 1805.04687.

Yu, Fisher, Yinda Zhang, Shuran Song, Ari Seff, and Jianxiong Xiao. 2015. "LSUN: Construction of a Large-scale Image Dataset using Deep Learning with Humans in the Loop." *CoRR.*

Yun, Sangdoo, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. 2019. "CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features." *2019 IEEE/CVF International Conference on Computer Vision (ICCV),* 6022–6031.

Zagoruyko, Sergey, and Nikos Komodakis. 2016. "Wide Residual Networks." In *BMVC.*

Zellers, Rowan, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. "HellaSwag: Can a Machine Really Finish Your Sentence?" In *ACL.*

Zendel, Oliver, Katrin Honauer, Markus Murschitz, Daniel Steininger, and Gustavo Fernandez Dominguez. 2018. "Wilddash-creating hazard-aware benchmarks." In *Proceedings of the European Conference on Computer Vision (ECCV),* 402–416.

Zhang, Hongyang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. 2019. "Theoretically Principled Trade-off between Robustness and Accuracy." In *ICML*.

Zhang, Hongyi, Moustapha Cissé, Yann Dauphin, and David Lopez-Paz. 2017. "mixup: Beyond Empirical Risk Minimization." *arXiv preprint arXiv:1710.09412*.

Zhang, Richard, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric." In *CVPR*.

Zhao, Hengshuang, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. 2017. "Pyramid Scene Parsing Network." *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6230–6239.

Zhong, Zhun, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2017. "Random Erasing Data Augmentation." *arXiv preprint arXiv:1708.04896*.

Zhou, Bolei, David Bau, Aude Oliva, and Antonio Torralba. 2019. "Interpreting Deep Visual Representations via Network Dissection." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41:2131–2145.

Zhou, Bolei, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. "Places: A 10 million Image Database for Scene Recognition." *PAMI*.

Zhou, Peng, Xintong Han, Vlad I Morariu, and Larry S Davis. 2018. "Learning rich features for image manipulation detection." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1053–1061.