MODEL ROBUSTNESS IN DATA-SCARCE REGIMES AND THE EFFECT OF FREQUENCY PERTURBATIONS

VERİ AZLIĞINDA MODEL GÜRBÜZLÜĞÜ VE FREKANS KARIŞTIRMANIN ETKİLERİ

MEHMET KERİM YÜCEL

PROF. DR. PINAR DUYGULU ŞAHİN
Supervisor
ASSISTANT PROF. RAMAZAN GÖKBERK CİNBİŞ
2nd Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
as a Partial Fulfillment to the Requirements
for the Award of the Degree of Doctor of Philosophy
in Computer Engineering

December 2022

ABSTRACT

MODEL ROBUSTNESS IN DATA-SCARCE REGIMES AND THE EFFECT OF FREQUENCY PERTURBATIONS

Mehmet Kerim Yücel
Doctor of Philosophy , Computer Engineering
Supervisor: Prof. Dr. Pınar Duygulu Şahin
2nd Supervisor: Assistant Prof. Ramazan Gökberk Cinbiş

December 2022, 146 pages

The last decade has witnessed the meteoric rise of data-driven methods, which has been elevated to new heights thanks to the availability of powerful hardware and abundant data. Despite their swift ascension, deep learning methods are repeatedly shown to have robustness problems; they can be tricked into making errors with minor changes in the input that are invisible to us humans, or they can not withstand certain failure modes common in real-life scenarios. This thesis focuses on the robust generalization problem, where two primary aims drive our research effort.

First, inspired from the surprising lack of thorough discussions on robust generalization in data-scarce regimes, we perform an exhaustive analyses on the robustness behaviour of models trained in zero-shot learning settings. We first show that discriminative zero-shot models have distinct robustness characteristics against adversaries, such as unseen and seen classes being affected disproportionately, the effect of original model accuracy and the stark differences between how zero-shot and generalized zero-shot accuracies degrade. We also identify the unique *pseudo-robustness effect* caused by adversaries, where models might be falsely declared as robust. We then extend our analyses to a more practical scenario, where images are corrupted with common image corruptions. We curate and present the first three datasets for corruption robustness analyses in the zero-shot literature. Using these datasets, we provide a set of rigorous analyses with a wider range of zero-shot models to assess their robustness against corruptions. Our results show that with key augmentation choices, we can improve the performance profiles of various models. Finally, we aggregate the results of adversarial and corruption robustness behaviours of zero-shot models and conclude with a thorough comparison.

Second, inspired by the fundamental techniques in image processing, we focus on using frequency-spectra information to improve model robustness. Assuming that the true label information of an image resides in its low-frequency components, we propose *HybridAugment* where images are augmented by randomly swapping their high-frequency component with other images. This augmentation is implemented in tandem with existing augmentations, and enforces the network to be less reliant on high-frequency information, which is a prime reason for model robustness issues. We then propose two variants of *HybridAugment*, where single or multiple image settings are used to perform the augmentation. With single and multi image augmentations being used at the same time, the results are further improved. Finally, inspired by the two orthogonal frequency-centric analyses (i.e. frequency bands and phase/amplitude decomposition) and the need to unify them, we propose *HybridAugment++* that performs a hierarchical augmentation in the frequency-spectra. In addition to swapping low and high-frequency components of images, *HybridAugment++* also swaps phase and amplitude of random images, but does so only on the low-frequency components. *HybridAugment++*, with its single and paired variants working in tandem, achieves state-of-the-art results in multiple benchmark datasets, showing its effectiveness.

Keywords: Robust Generalization, Zero-Shot Learning, Frequency Analyses, Data Augmentation, Image Recognition

ÖZET

VERİ AZLIĞINDA MODEL GÜRBÜZLÜĞÜ VE FREKANS KARIŞTIRMANIN ETKİLERİ

Mehmet Kerim Yücel
Doktora, Bilgisayar Mühendisliği
Danışman: Prof. Dr. Pınar Duygulu Şahin
Eş Danışman: Asistan Prof. Ramazan Gökberk Cinbiş
Aralik 2022, 146 sayfa

Geçtiğimiz on yıl, veri bazlı metodların çeşitli disiplinlerde yükselişine tanık olmuştur. Etkileyici yükselişlerine rağmen, derin öğrenme metodlarının gürbüzlük problemlerine yatkın oldukları gözlemlenmiştir; bu modellerin, insanların farkedemeyeceği şekilde değiştirilen resimlerde hatalı tahmin yaptıkları, hatta günlük senaryolarda olağan olarak gerçekleşen ve girdileri etkileyebilen olaylar karşısında çalışamadıkları gözlemlenmiştir. Bu tez, derin öğrenme modellerinin, gürbüz bir şekilde genellenmelerine odaklanmaktadır ve bu konuda özellikle iki tane amaca yoğunlaşmıştır.

Veri azlığının olduğu senaryolarda eğitilmiş modellerin gürbüzlüğüne dair literatürde çalışma olmamasından esinlenerek, bu tezde ilk olarak *sıfir-atış* senaryolarda eğitilen modellerin gürbüzlük karakteristikleri incelenmiştir. İlk olarak, ayrımcı sıfır-atış modellerin, *düşmancıl* resimlere karşı farklı gürbüzlük özelliklerine sahip oldukları saptanmıştır; görülmüş ve görülmemiş sınıfların farklı etkilenmesi, asıl model başarımının etkisi, sıfır-atış ve genelleştirilmiş sıfır-atış senaryolarının çok farklı etkilenmesi, bu davranışlara bazı örneklerdendir. Bundan sonra, sadece düşmancıl resimlere karşı görülen *sözde gürbüzlük* etkisi saptanmıştır ve analiz edilmiştir; bu etki, aslında gürbüz olmayan modellerin gürbüz olarak algılanmasına yol açabilmektedir. Bundan sonra, analizlerimiz daha pratik bir senaryo olan *olağan resim bozulmalarına* odaklanmıştır. Öncelikle, bu senaryolarda analiz yapabilmek için, sıfır-atış modellerinde olağan resim bozulmalarını analiz etmeye yarayacak, literatürde önceden örneği olmayan, üç veri seti hazırlanmıştır. Bu veri setleri kullanılarak, ve kullanılan sıfır-atış modelleri çeşitlendirilerek, olağan resim bozulmalarına karşı analizler yapılmıştır. Çeşitli veri büyütme tekniklerinin, var olan sıfır-atış modellerinin sonuçlarını iyileştirdiği görülmüştür. Son olarak, düşmancıl ve olağan bozulma analizlerin sonuçları karşılaştırılmış ve sonuçlar verilmiştir.

Temel görüntü işleme tekniklerinden olan resim frekans analizi metodlarından ilham alınarak, resimlerin frekans bilgilerinin model gürbüzlüğünü geliştirme ihtimalleri araştırılmıştır. Resimlerin asıl önemli olan özelliklerinin düşük frekanslarda olduğunu baz alarak, yeni bir veri büyütme tekniği geliştirilmiştir. *HybridAugment* adını verdiğimiz teknik, resimlerin yüksek ve düşük frekans bileşenlerinin rastgele bir şekilde değiştirilmesiyle yapılmaktadır. Başka veri büyütme teknikleriyle de çalışabilen bu metod, modellerin öğrenme sürecinde düşük frekans bileşenlerine yoğunlaşmasını sağlamakta, ve gürbüzlük problemlerinin sebebi olarak gösterilen yüksek frekans bileşenlerine yoğunlaşmalarını azaltmaktadır. *HybridAugment* metodunun iki versiyonu sunulmuştur; tek ve çoklu resimlerlerle çalışan bu versiyonlar, birlikte de çalışabilmekte ve sonuçları daha da iyileştirmektedir. Son olarak, resimlerin faz bileşenenin daha ziyade uzaysal bilgiye sahip olduklarını baz alarak, *HybridAugment++* metodu geliştirilmiştir. Bu metod, resimlerin faz ve büyüklük bileşenlerin rastgele değiştirilmesi ile *HybridAugment* tekniğini hiyerarşik olarak uygulamaktadır. *HybridAugment++*, tekli ve çoklu versiyonları aynı anda çalıştırıldığı zaman, birden fazla veri setinde literatürdeki en iyi sonuçları elde etmektedir.

Anahtar Kelimeler: Gürbüz Genelleme, Sıfır-Atış Öğrenme, Frekans Analizi, Veri Büyütme, Resim Tanıma

ACKNOWLEDGEMENTS

First and foremost, I extend my eternal gratitude to my advisors Prof. Pinar Duygulu Sahin and Prof. Gokberk Cinbis for taking a chance with me. Their endless patience, support and guidance have made this thesis happen and I feel lucky to have had the chance to be their mentee. From the calm shores of ideation to the bloody trenches of coding and manuscript preparation, they have always been there, and I will be eternally grateful.

I would like to thank my thesis committee members Prof. Erkut Erdem and Prof. Emre Akbas for bearing with me throughout multiple committees and providing invaluable feedback which improved this thesis to its core. Their encouragement has been an integral part of this research effort. I would also like to thank the jury members for their invaluable feedback.

This thesis has been the culmination of a really long journey, even longer than what it took to finish my PhD degree. I would like to pay homage to some of the key people who have shaped my professional career; Prof. Abdul Sadka for introducing me to computer vision, Dr. Mathew Legg and Dr. Vassilios Kappatos for bringing out the researcher in me, and Dr. Onay Urfalioglu and Dr. Berker Logoglu for teaching me how to be a good leader.

All my current or former colleagues deserve a heartfelt thank you; they have played key parts in this research endeavour, one way or another. From the freezing nights of Ankara to intense gunfights in Call of Duty, I thank all my friends for their understanding and support. I owe much to my family, from my sisters and father to my lovely in-laws, I thank them for being there. I pay special homage to my sister Dr. Ela Burcu Ucel, who has been my role model in academia as well as in mental stability. To my lovely mom; it is an honour to be receiving my PhD degree from your alma mater some 47 years ago after you. Time is cruel and it is saddening to realize it has been 17 years already. I will always miss you and love you.

Finally, I dedicate this thesis to my lovely wife Gökçe. I know that she knows I love her, but no matter how much I say it, it never feels enough. Thinking of the support, love and even the paper feedbacks she provided, it is only just to grant this degree to her and not me.

CONTENTS

Page

TABLES

Page

FIGURES

Page

ABBREVIATIONS

ML:	**M**achine **L**earning
CNN:	**C**onvolutional **N**eural **N**etwork
SGD :	**S**tochastic **G**radient **D**escent
ZSL :	**Z**ero-**S**hot **L**earning
GZSL :	**G**eneralized **Z**ero-**S**hot **L**earning
FFT :	**F**ast **F**ourier **T**ransform
IFT :	**I**nverse **F**ourier **T**ransform
CE :	**C**orruption **E**rror
mCE :	**M**ean **C**orruption **E**rror
AUROC :	**A**rea **U**nder the **R**eceiver **O**perating **C**haracteristics

1. INTRODUCTION

In this chapter, we motivate our research, present a high-level overview of our research and contributions, and finally provide a detailed organization of the thesis.

1.1. Background and Motivation

The goal of imitating the complex human vision system has been a centre of attention for the research community, which eventually gave birth to the 60+ year old field of computer vision. The computer vision journey is widely believed to have started in 1957 with the very first digitization of an image, led by scientists of American National Institute of Standards an Technology (ANIST) [12]. Widely accepted as the *father* of computer vision, Lawrence Roberts was one of the first (if not *the* first) scientists who explored the acquisition of 3D information from a 2D image in his PhD thesis in MIT [13]. These seminal works have been succeeded with a plethora of equally impactful works, from Neocognitron [14] to Viola-Jones [15], SIFT [16] to LeNet [17], and countless others that are being published even to this date.

Due to the high-dimensional and complex nature of images, the earlier computer vision methods often struggled to make real-world impact, primarily due to their hand-crafted nature that only worked under strict, simplifying assumptions. The natural complexity of images often calls for an automated discovery of the patterns inherent to such images. Therefore, computer vision research has extensively utilised the advances in statistical/machine learning, such as SVMs [18], CNNs [17], graphical models [19], random forests [20], and many more. Despite the increasing use of machine learning methods in computer vision, the features that represent images have been primarily hand-crafted; powerful representations such as SIFT [16], SURF [21] and ORB [22] have found use in various vision tasks.

Much of the core computer vision research has stayed the same throughout the years and primarily focused on one thing; *effectively and efficiently representing images*. A breakthrough moment arrived in 2012 with AlexNet [23], where a combination of powerful

GPU hardware, deeper convolutional neural networks and new regularization tricks showed that it is quite possible to *learn* powerful image representations from data, rather than handcrafting these representations. AlexNet started an AI storm that disrupted not only vision, but any other discipline that can *scale* with data. Architectural novelties such as VGG [24], ResNet [4], transformers [25], and countless other advances such as GANs [26], knowledge distillation [27], self-supervised learning [28] and diffusion models [29] in what is now called *deep learning* took computer vision to new heights that were unimaginable a mere decade ago. The fact that ResNet paper got over 140000 citations is a testament to its impact, and also to the ever-growing size of the field.

> *’All that glitters in not gold.’*
>
> *William Shakespeare*

Following the meteoric rise of deep learning in computer vision and other disciplines, an *intriguing* study published in 2014 [30] showed a troubling pattern; these powerful neural networks can be *fooled* to misclassify images via the addition of imperceptible (to the human eye) noise, or *perturbations*, to the images. Essentially, an imperceptible noise added to images is shown to be capable of *completely invalidating* otherwise state-of-the-art methods. This problem is also shown to be present for numerous computer vision solutions/tasks, and even other modalities such as speech [31] and natural language [32]. This vulnerability of modern neural networks started an *arms race* between adversarial attacks and defenses, where community rushed into finding methods to *patch* their networks for protection or created new methods to *fool* them. These developments in adversarial machine learning inspired researchers to focus on the *bigger picture*, namely the *robust generalization* of neural networks, which covers analyses in many other robustness venues, such as out-of-distribution robustness [33] and robustness to common image corruptions [9].

There has been an increase in interest on robust generalization in machine learning, especially since real-world applications of these models require verifiable robustness against malicious perturbations, or common types of corruptions an image can be exposed to in real-life systems. A wide range of literature on the topic is available, such as considering the issue from a simple generalization perspective [34] to identifying robust/non-robust features [35], from designing robustifying data augmentations [36] to analysing robustness from a frequency-spectrum perspective [37].

Although the field of robust generalization is quite active, there are areas which have been largely neglected and unexplored. One such area is the effect of *the level of supervision* on robust generalization; virtually all papers in the literature somewhat assume that models are trained under a fully supervised setting, where labels are available for each sample. The availability of labels is a simplifying assumption, as label acquisition is often expensive, if possible at all. In its extreme, the availability of training samples for certain classes may not even be possible, leading to severe class imbalance. Zero-Shot Learning and Generalized Zero-Shot Learning [7] lie in such extremes, where training data is available for only a subset of classes, but models are evaluated on their ability to perform also on classes unseen during the training. The current literature fails to answer these questions, especially in such extremes.

Another area of interest is the development of data augmentation methods to increase model robustness. Stemming from the perspective that considers robustness as a generalization issue, many methods diversify training data distribution in such a way that models become more robust, ideally while keeping their accuracies on uncorrupted data at least the same. The majority of the studies focus on learning such augmentations [38] or use a combination of existing, common image transformations [36]. There is an emerging and relatively unexplored branch of work that focuses on the frequency spectra of images for robustness [37]. As different frequency bands of images are known to carry different information, their individual effects on robustness is actively being explored by the community. However, their use for formulating robustifying data augmentation methods has been largely neglected.

In this thesis, in line with the previous discussions, we primarily focus on model robustness in scarce data regimes, and then develop methods for model robustification. Essentially, we first ask then answer the following questions.

- **Q1:** *What are the robustness characteristics of models trained with heavily imbalanced data; i.e. in zero-shot settings? Are their robustness characteristics any different from their fully supervised counterparts? If so, how and why?*
- **Q2:** *Can we develop data augmentation methods that would produce robust models? Specifically, can we leverage frequency spectra information of images to achieve this goal? If so, how?*

Throughout this thesis, we answer these two questions through theoretical discussions, rigorous experimentation and detailed analyses. The answers we provide in this thesis, along with relevant key insights, shed light on model robustness in data-scarce training regimes, specifically for zero-shot settings. Furthermore, the methods presented in this thesis are shown to be effective and superior to current methods on multiple benchmarks datasets related to model robustness.

1.2. Contributions

This thesis reports three primary contributions, each of which are explained in their respective chapters. The first two chapters aim to collectively answer the first question (**Q1**), whereas the third chapter focuses on the second question (**Q2**) and presents multiple methods which answer it.

Chapter 3. Having observed the negligence of robustness analyses for non-supervised methods in the literature, we focus our efforts on models with weak-to-no supervision. Specifically, we choose discriminative Zero-Shot Learning (ZSL) methods as they reside on one extreme of data imbalance (where for some classes no training samples are available), among other reasons detailed in the relevant chapter. We identify adversarial performance of the model as the first key venue of robustness, and choose a family of representative (discriminative) ZSL models. We subject these models to a multitude of common adversarial attacks and defenses, and record their performance. Their performance is then analysed from multiple key aspects; such as the effect of training data (i.e. per-class sample count, number of classes, etc.), the trends in class boundary transitions (i.e. class transitions from false to correct, class transitions from unseen to seen, etc.), the potentially adverse effects of defenses, the effect of model *maturity* (i.e. original accuracy of the model) and the discrepancy between how ZSL and GZSL performance are affected. We show and discuss that data-imbalanced models have different characteristics compared to fully supervised models, when exposed to adversaries. Some exemplary differences are the discrepancy between unseen and seen class performance degradation and the severe effect of model maturity. We also identify the *pseudo-robustness effect* often observed in our analyses, where absolute metrics may not always reflect the robustness behaviour of the model.

Chapter 4. We continue our work on the extensive analyses presented in Chapter 3. Although adversarial robustness is arguably *the venue* that sparked the robust generalization discussion in modern ML literature, they exhibit specific characteristics that lowers the *practicality* of their threat; adversaries are *worst-case* scenarios specifically crafted with malicious intent. Although quite possible, such incidents are not *that* likely to occur. With this motivation, we search for a more *practical* venue of robustness analyses, with a more immediate effect of real-world applications. We define *common image corruptions* as our new robustness front, where images undergo effects that are common in real-world, such as digital artefacts (i.e. jpeg compression, noise), various blur types and weather effects. Such effects, unlike adversaries, are not worst-case scenarios (i.e. carefully crafted by experts to be imperceptible) and thus have a higher threat practicality. Since there are no benchmark datasets to analyse corruption robustness of ZSL models, we present not one but three datasets that fill this gap. We present AWA2-C, SUN-C and CUB-C, which are corrupted versions of existing ZSL datasets. Using these datasets with a multitude of methods that claim to improve corruption robustness in fully supervised settings, we perform an exhaustive experimentation and provide similar analyses to those of Chapter 3.; the effect of training data (i.e. per-class sample count, number of classes, etc.), the trends in class boundary transitions (i.e. class transitions from false to correct, class transitions from unseen to seen, etc.), the effects of defense methods, the effect of model *maturity* (i.e. original accuracy of the model) and the discrepancy between how ZSL and GZSL performance are effected.

Furthermore, we expand our selection of ZSL models to answer a simple question; *do our findings generalize to a larger family of ZSL models?*. We perform the same experiments with other models, specifically chosen to be more accurate than our primary selection, and show that our insights hold for a multitude of ZSL models, and not just a single family of them. Our results show, among other things, that some defense methods actually improve the accuracy of these ZSL models and set new strong baselines. We also show and discuss that data-imbalanced models have different characteristics compared to fully supervised models, when exposed to corruptions. Finally, we combine the findings of Chapters 3. and 4. to present a set of rigorous comparative analyses to highlight the differences between the effects of corruptions and adversaries, and present a detailed roadmap/starting point for further studies in the field.

Chapter 5. Finally, we focus on developing methods that improve the robustness of models, regardless of the supervision profile they are trained under. Inspired by the recent studies that are based on frequency spectra to explain model robustness [10, 35, 37, 39], and keeping in mind that the most successful robustness improvement methods do so by introducing data augmentations, we focus on developing a frequency spectra based data augmentation method. To this end, we propose *HybridAugment*, which closely follows the well-known hybrid images [40] that combine high-frequency and low-frequency content of images. The core idea behind our method is simple; we hypothesize that the *true* label information of an image primarily reside in the low-frequencies of the image. Furthermore, it is discussed that Convolutional Neural Network often use high-frequency content more than us humans [39], which leads to robustness issues. Therefore, *HybridAugment* aims to force the networks to rely on low-frequency information rather than high-frequency information present in the data, thus improving their robustness. We propose two variants; *HybridAugment-Paired* and *HybridAugment-Single*, where we use two random images and the different transformations of the same image to provide low and frequency content for the HybridAugment, respectively. Furthermore, we show that their combination, *HybridAugment-PairedSingle*, outperform both of these two variants. We show that HybridAugment-PairedSingle produces highly competitive results in several benchmark datasets.

We then identify arguably two of the most prominent frequency-centric analyses in the literature; frequency-bands (i.e. low vs high) and phase/amplitude decomposition. Motivated by the fact that *HybridAugment* shows that the former works, and the latter is shown to work by the research community, we unify them into a single, hierarchical augmentation method we call *HybridAugment++*. *HybridAugment++* first separates the low and high frequency content of an image, and since we know that the label information is primarily in the low-frequency, performs the amplitude/phase swap on the low-frequency content. It then combines the augmented low-frequency content with the high-frequency content of another image to generate the augmented image. *HybridAugment++* has single and paired variants called *HybridAugment++ Paired*, *HybridAugment++ Single* and their combinations *HybridAugment++ PairedSingle*. Our results on multiple benchmark datasets show that *HybridAugment++ Single* outperforms other single-image augmentations on multiple CNN architectures. Similarly, *HybridAugment++ Paired* comfortably outperforms other multi-image augmentations on the same CNN architectures. Finally, their combination *HybridAugment++ PairedSingle* achieve state-of-the-art robustness results on multiple datasets with a significant margin.

Our contributions, in a more compact way, are listed below.

- We present a thorough adversarial robustness analyses of discriminative ZSL models, and show the effect of extreme data imbalance, as well as model accuracy on adversarial robustness. We also identify the *pseudo-robustness effect* with adversaries.

- We present three ZSL datasets for image corruption robustness analyses. To the best of our knowledge, there is no other dataset presenting a testbed for ZSL corruption robustness analyses.

- We present a thorough common image corruption analyses of discriminative ZSL models, and show the effect of extreme data imbalance and model accuracy. We expand our analyses to other families of ZSL models, and show that our insights hold for a broader range of model families.

- We perform a thorough comparison of adversarial and corruption robustness behaviour of discriminative ZSL models, show the key differences and present roadmaps for future research.

- We present new strong baselines for various existing ZSL methods by using robustness-enhancing data augmentation methods.

- We present *HybridAugment*, a frequency-spectra based data augmentation method that improves model robustness across multiple datasets. We present three variants of *HybridAugment*.

- We present an enhanced version of *HybridAugment*, called *HybridAugment*++, that achieves state-of-the-art robustness metrics on multiple benchmark datasets. We present three variants of *HybridAugment*++.

1.3. Organization

The organization of the thesis is as follows:

- Chapter 1 presents our motivation, lists our contributions and outlines the scope of the thesis.

- Chapter 2 provides the relevant foundational information and literature review on Zero-Shot Learning, Robust Generalization and Frequency-Spectra in Images for better comprehension of the thesis.

- Chapter 3 presents the adversarial robustness analyses of discriminative ZSL models and identifies important phenomena such as the *pseudo-robustness* effect.

- Chapter 4 builds on Chapter 3 and presents the corruptions robustness analyses of a larger family of ZSL models, performs a rigorous comparison between adversarial and corruption robustness of discriminative ZSL models and presents three new datasets.

- Chapter 5 introduces *Hybrid-Augment* and *HybridAugment++* methods that improve robustness across multiple datasets and multiple network architectures.
- Chapter 6 presents the summary of the thesis, discusses the highlights the thesis and presents potential future directions of research.

The outcomes of this theses are reported in several publications and correspoding software packages; one conference paper, one journal paper and a working paper to be submitted to a conference or a journal.

Paper 1: *Yucel, Mehmet Kerim, Ramazan Gokberk Cinbis, and Pinar Duygulu. ”A deep dive into adversarial robustness in zero-shot learning.” European Conference on Computer Vision Workshops. Springer, Cham, 2020.*. This paper largely corresponds to what is reported in Chapter 3., and presents the adversarial robustness analyses of discriminative ZSL models.

Paper 2: *Yucel, Mehmet Kerim, Ramazan Gokberk Cinbis, and Pinar Duygulu. ”How robust are discriminatively trained zero-shot learning models?.” Image and Vision Computing 119 (2022): 104392.*. This paper largely corresponds to what is reported in Chapter 4., and presents the corruption robustness analyses of discriminative ZSL models as well as three new datasets we propose.

Software 1: The code used to perform the experiments presented in **Paper 1** can be found at `https://github.com/MKYucel/adversarial_robustness_zsl`.

Software 2: The code used to perform the experiments presented in **Paper 2**, as well as our proposed datasets AWA-C, SUN-C and CUB-C can be found at the link given below `https://github.com/MKYucel/zero_shot_corruption_benchmarks`.

Working Paper: The contents of Chapter 5 will be compiled into a paper for submission.

Working Paper Software: The code used to perform the experiments presented in Chapter 5 will be made available at `https://github.com/MKYucel/hybrid_augment`.

2. RELATED WORK

In this chapter, we present the relevant foundational information we believe to be required for a better comprehension of this thesis. We will provide a brief overview of the basics, and then move forward with relevant literature.

2.1. Fundamentals

First, we cover the fundamentals; we talk about the core learning mechanisms in ML, different supervision profiles in the current ML literature and introduce commonly used CNN architectures.

2.1.1. A primer on Machine Learning

Machine Learning. Machine learning has many definitions, but we consider it to be the discipline of learning from observations (data) to approximate functions and discover patterns in the data. It is especially useful in complex cases, where a function does not have a closed form solution or even can not be described mathematically. Imagine a scenario where we are tasked with recognizing a cat in an image. The question is simple; *how can we mathematically describe a cat?* We do have an idea about what a cat is; it has ears, whiskers, mouth, eyes and paws. However, it is nearly impossible to *formulate* a cat mathematically as cats differ in size, breed, age, gender and images in cats with them differ in viewpoint, scale, color, illumination, etc. In this scenario, the best we can do is to *approximate a function* that recognizes (and implicitly models) a cat; it should be able to say whether a query image has cats or not. Such a scenario is ideal for machine learning; i) we have lots of observations (i.e. images) of cats and ii) we know they are cats (i.e. labels). The rest depends on the algorithmic choices we adopt. Machine Learning is built on a multi-disciplinary paradigm, where it leverages mathematics (i.e. optimization, algebra, probability), statistics (i.e. learning theory), psychology and computer science. It has been around for quite a while, but especially in the recent years it has become too big of a topic to summarize in a mere chapter. Here, we describe the relevant basic building blocks for a brief overview.

Perceptron. Machine Learning borrows quite a lot from statistics and learning theory. Along with fundamental methods like logistic regression, kernel methods, clustering and dimensionality reduction, nearest neighbors and the like, the field has taken a lot of inspiration from biology, especially from a learning mechanism perspective. The basic processing unit of the brain is called *neuron*, which essentially receives and forwards electrical information to other neurons with connections called *synapses*. Synapses have *weights* which corresponds to the strength of the connection, and the output of a neuron is essentially a combination of all input signals weighted by individual synaptic weights. As a rough imitation of neurons, the seminal perceptron [41] classification model $f(\cdot; W)$ was proposed where

$$f(X; W) = \sigma\left(b + \sum_{i=1}^{N} W_i \cdot X_i\right) \tag{1}$$

where X is the input data, N is the number of inputs (i.e. dimension of the input), σ is the step function, W_i is the i^{th} connection (i.e. synaptic) weights, X_i is the i^{th} input and b is the bias term. The primary problem with perceptron is that it provides a linear decision boundary despite the non-linearity σ; such limitation would lead to insufficient discriminatory performance in complex, real-world data. A natural solution to that is the famous *multi-layer perceptron*, where a number of perceptron units are stacked together to form a *deeper* classifier able to handle non-linear decision boundaries. Note that Equation 1 is the widely used fully-connected layer in modern neural networks, where activation functions might differ based on network design.

Loss Functions. Now that we have a design *theoretically* capable of approximating any function, we need a *way* to approximate any function. Before going into the learning algorithms, we need a way to measure how well our method is doing on given observations. This is called a *loss function* (i.e. cost function, objective function) and they map the values estimated by the model to the target values, and produce a value that computes the error between the two. Depending on the task, this loss function is either minimized or maximized. The space of loss functions is extremely large, spanning cross-entropy to KL-divergence [42], smooth-L1 [43] to gradient losses [44], all the way up to *learned* loss functions such as discriminator networks [26]. An example loss often used in our thesis is the cross-entropy (CE) loss that measures the performance of a classification model given a target probability distribution (i.e. labels), and its value decreases as the prediction distribution and target distribution converge. It is defined as

$$CE = -\sum_{c=1}^{M} t_i log(s_i) \tag{2}$$

where M is the number of classes, t_i is the ground-truth and s_i is the predicted class for class i. Note that s_i are the softmax-normalized output probabilities.

Gradient-based Learning. We now have a function approximator and an evaluation metric, but still lack the mechanism to update our function approximator such that it *learns* to, well, approximate the function. The basic idea for this learning/optimization problem is defined as

$$W' = \underset{W}{\operatorname{argmin}} \sum_{n=1}^{N} (Loss(y_n, p_n)) \tag{3}$$

where the aim is to find a set of weights W' that parameterizes the function approximator $f(\cdot; W)$, by minimizing the $loss$ between ground-truth labels y and model predictions p over N number of samples.

Actually performing Equation 3 requires two ingredients; i) an optimization method to find the necessary parameter updates and ii) a way to do this for all parameters in the network. Among the vast optimization literature, the most commonly used optimization method in modern ML (and in our thesis) is the *gradient descent* method [41] that aims to find a local minimum of our function approximator $f(\cdot; W)$. Assuming that $f(\cdot; W)$ is an end-to-end differentiable single layer model, gradient descent first calculates the gradient of the loss function with respect to the parameters W of $f(\cdot; W)$, and uses this gradient to update the parameters W. It is formulated as

$$W_{t+1} = W^t - \eta \frac{\partial Loss}{\partial W^t} \tag{4}$$

where t indexes iteration/time, ∂ denotes the partial derivative operation and η is the learning rate parameter. The final block of the puzzle is simple; *how do we do this for a network with many layers*? The answer is the chain-rule from algebra, which leads to the *backpropagation* [45] algorithm. Assuming an end-to-end differentiable, multi-layer model $f(\cdot; W)$, the backpropagation algorithm calculates the necessary gradient updates (required by the gradient descent algorithm) for all model layers in a backward fashion, starting from the final layer of the network. For example, to calculate the parameter update of the i^{th} layer of the network, backpropagation does the following

$$\frac{\partial Loss}{\partial W_i} = \frac{\partial Loss}{\partial Z_L} \cdot \frac{\partial Z_L}{\partial Z_{L-1}} \cdots \frac{\partial Z_i}{\partial W_i} \tag{5}$$

where L is the number of the layers in the network and Z_i refers to representations (i.e. hidden layer outputs) of the i^{th} layer.

There are additional mechanisms, such as learning rates (see Equation 3), weight decays etc. which are controlled by the so-called *optimizers* during the training; ADAM is a prime example of these optimizers [46]. Furthermore, since all data samples might not fit to the machine memory during training, in each iteration of Equation 3 we often randomly sample a subset of the N training samples (i.e. batch) and perform the training; this process is called *stochastic gradient descent*. We sample necessary number of batches until we cover the entire dataset (i.e. epoch), and then train as much as we like. In short, the process can be summarized as follows.

- We randomly sample a batch from the training samples.

- We feed this batch to our network $f(\cdot; W)$ and get our predictions.
- We calculate the loss value $Loss$, perform backpropagation (Equation 5) to find the gradient updates for all layers.
- We perform the necessary gradient updates (Equation 4) with the help of the optimizer.
- We repeat until convergence, or until we reach a certain number of iterations/epochs.

2.1.2. Supervision Profiles in Machine Learning

As explained in Chapter 2.1., ML methods require large sets of data to model complex tasks. Furthermore, these large datasets should come with label information; i.e. such as the class label of the image (i.e. cat image, dog image, etc). This leads to an expensive and laborious process, where manual annotations for many images need to be produced. Note that this problem becomes even harder, nearly impossible with dense annotations; imagine a segmentation mask annotation where every pixel must be annotated with class labels. This will be extremely time consuming for a human annotator, thus quite expensive. Luckily, machine learning is not formed of just supervised methods. We now briefly cover popular supervision profiles.

Supervised Learning. Often called *fully-supervised* learning, supervised learning methods are already mentioned in this thesis; when we have a dataset and associated ground-truth labels, we can leverage such methods to achieve our goal. Since the availability of labels provide a strong supervisory signal for learning, such methods often achieve state-of-the-art in many tasks. Therefore, many bleeding-edge methods in various disciplines leverage fully-supervised learning.

Unsupervised Learning. Unsupervised methods' primary aim is to discover hidden structures present in the data, without *any* external supervisory signal. Various clustering or dimensionality reduction methods are such examples [47, 48]. Due to the lack of a supervisory signal, these methods often struggle to reach bleeding-edge results. However, eliminating the need of labels make them quite desirable due to lowered costs.

Semi-supervised learning. In semi-supervised learning, methods make us of label and unlabeled data to further achieve good results. By definition, it combines supervised and unsupervised learning; i.e. we can leverage unlabeled data to learn good representations and then finetune the model on labeled data to improve the results. Alternatively, we can use a pretrained model to *pseudo-label* an unlabeled dataset, and then train another model on these generated labels. Semi-supervised learning is a broad term and may refer to other approaches as well, such as weakly-supervised methods where the annotations are not of the same structure as the model predictions; weakly-supervised object detection methods without localization annotations [49, 50] or stacked networks with implicit localization blocks [51] are such examples.

Self-supervised methods. Self-supervised methods are quite similar to unsupervised methods, but with key differences. In self-supervised methods, as the name implies, we *craft* our own supervision, either by domain knowledge, data design or designing pretext tasks. An example of using domain knowledge is the self-supervised depth estimation literature [52], where left-right image consistency is used as a reconstruction loss by warping one image onto another. Here, although there are no ground-truth labels present, we use geometric principles to derive a supervisory signal. Data design refers to data engineering to generate ground-truth labels; a prime example is video frame interpolation where one can simply hold-out frames of a video to reduce the frame rate, and teach the model to predict the held-out frames. Pretext-task based self-supervision comes in the form of pretraining; models are shown to learn useful representations when they are forced to solve jigsaw puzzles [53].

Other supervision profiles. We note that this literature is vast, and covering all examples is not tractable. However, aside from dataset-level supervision profiles, there are other profiles where per-class sample count is of concern. Few-shot learning methods [54] aim to learn from only a few images, whereas zero-shot learning methods [55] use *absolutely no* data (labeled or unlabeled) for specific classes in the test distribution. Note that ZSL is a pillar of our research and has a dedicated section (see Chapter 2.2.).

2.1.3. Convolutional Neural Networks

Convolution Layer. With the seminal works of LeNet [17] and AlexNet [24], CNNs have sparked the deep learning revolution. So far, we mentioned the fully-connected (FC) layers (Equation 1) and networks formed of these layers. However, in high-dimensional data, using FC layers quickly become intractable due to extreme number of parameters as they are densely connected. Inspired from human visual systems and algebra, convolution layers use the local convolution operation. A convolution operation is performed using a convolution kernel (also called a *filter*), where we spatially slide and compute the dot products between this kernel and image patches. After the multiplication (i.e. essentially a Hadamard product), we sum the resulting matrix and produce a value. Note that if the size of the overall image is bigger than the convolution kernel (which is often the case), we will end up with a matrix of resulting values; this is called an *activation map* (i.e. feature, representation). The size of each activation maps depend on convolution kernel size, stride and the input image size.

The convolution layers have several advantages; they i) are local in nature, which aligns well with image modality due to closely-spaced pixel relations, ii) perform parameter sharing spatially; i.e. same kernel is used throughout the whole image and iii) are sparsely connected and save precious memory/storage/runtime. An example is shown in Figure 2.1. Note that CNNs use convolution layers, but might use other layers as well; such as fully-connected layers, pooling layers, non-linear activations, normalization layers, regularization layers (i.e. dropout),etc. We do not mention much of these layers for brevity.

Architectural Advances. The literature on CNNs is quite vast; what [17] started had huge consequences, and the end is not in sight yet. Here, we briefly review some CNN architectures. Note that these architectures are reviewed primarily because we extensively use them in Chapter 5.

AllConv [56] network removes the non-convolutional layers from CNNs (except the non-linearities), and replaces the pooling layers with a convolutional layer with increased stride. Note that the FC-layers can also be replaced by 1×1 convolution layers, leading

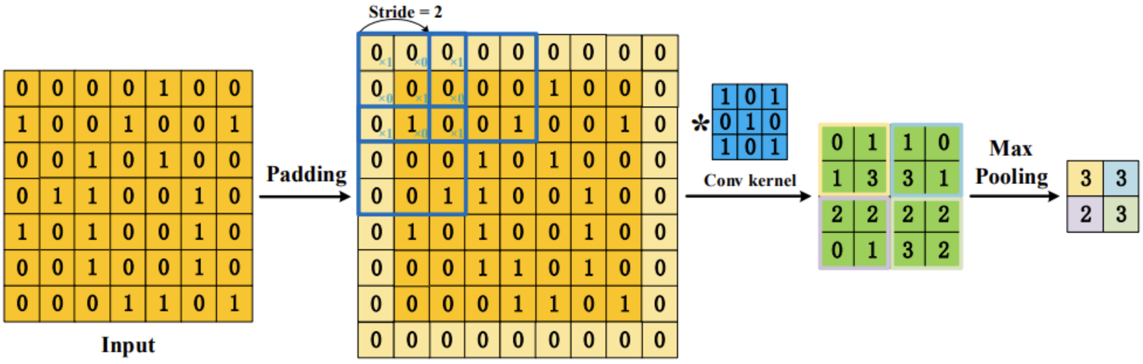


Figure 2.1 A diagram of the convolution operation. Image credit: [3].

to an All-Convolution (AllConv) network. AllConv is a fundamental CNN architecture that showed the power of convolutional layers, and the fact that they can be the sole building block of advances architectures.

ResNet [4] architecture is arguably the most impactful CNN architecture in the literature, where it is still the go-to architecture for virtually all vision tasks. ResNets ask a simple question; *how can we train even deeper networks*? The problem before ResNets was the the inability to train deeper models due to the vanishing gradients problem, where gradients would be so small to provide meaningful parameter updates to earlier layers of a CNN. By introducing the identity residual connections (i.e. a special case of another seminal paper Highway Nets [57]), authors showed that even a 150-layer deep network can be trained successfully with even better accuracy, thanks to the significantly improved gradient flow. The residual block that forms ResNets is shown in Figure 2.2.

WideResNet [58] thoroughly discusses the effects of width and depth in residual networks, and then propose a wider version of it. Specifically, a wider ResNet block is proposed. WideResNets are known to be useful especially in terms of computational performance, as the sequential nature of the network (i.e. depth) is traded against the number of filters (i.e. width), which theoretically keeps a similar presentation power. Increasing the width increases the computational performance much more gracefully compared to increasing the depth, as more filters make better use of GPU parallelism.

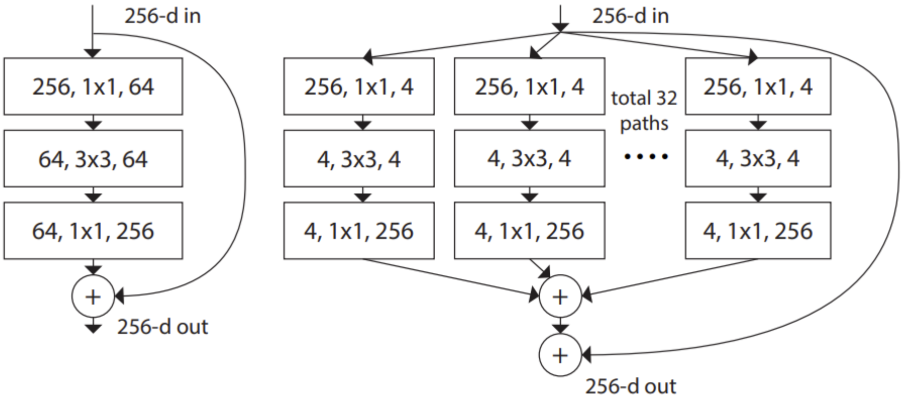


Figure 2.2 The ResNet [4] and ResNext [5] blocks. Image credit: [5].

ResNeXt architecture builds on the ResNet block and introduces the hyperparameter *cardinality* which refers to the number of branches available in the residual block. Essentially, multiple branches are fed the same input tensor where the output of each branch is aggregated at the end, providing the output of the so-called ResNeXt block. Note that the cardinality is similar to the width, except *width* refers to number of filters in a layer whereas *cardinality* refers to number of parallel branches in a block. ResNeXt block is shown in Figure 2.2.

DenseNet [6] explores the extreme where every layer is connected to every other layer. Essentially, each layer takes the previous feature maps as input. According to the authors, DenseNet addresses the vanishing gradient problem, strengthen feature propagation and promote feature reuse. DenseNet is formed of DenseBlocks, where such dense connections are established between layers with same feature dimensions. DenseNet produces strong results, at the cost of degraded computational performance. DenseNet block is shown in Figure 2.3.

2.2. (Generalized) Zero-Shot Learning

In this section, we focus on the relevant information on Zero-Shot and Generalized Zero-Shot Learning. We first motivate why we need them, formally formulate the challenges and then present a review of methods relevant to our thesis.

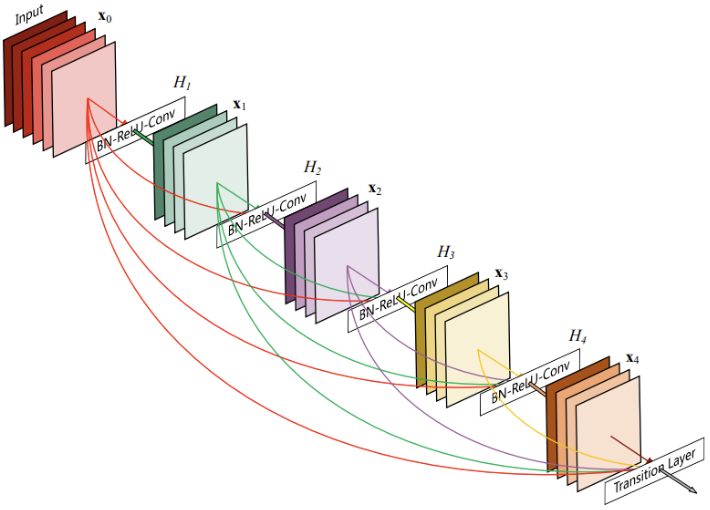


Figure 2.3 The DenseNet [6] block. Image credit: [6].

2.2.1. Preliminaries.

As discussed in Chapter 2.1.2., although we have systems that can scale their performance with data, finding the data and annotating them is an important bottleneck. Considering a simple image classification problem, we must know that not every class is created equally; some classes will have more samples than others (i.e. more images of cats on the internet than, well, *anything else*). One-shot and few-shot learning methods are specifically devised to address this class imbalance issue. What happens when we go the extreme, and have *no* samples of some classes at all?

The first answer that comes to mind is that we can not recognize them at all. Specifically, our classifier will assign a different prediction label to these samples, leading to completely incorrect results. Detecting such samples are quite important in real-world applications. Open Set methods and out-of-distribution detection methods tackle this issue up to a degree, but they can only detect that such samples are not of the training classes. Essentially, all these methods are limited to recognizing the classes they are trained on. We then revise our question; how do we detect *and recognize* the classes not used in the training?

Zero-Shot Learning. Zero-Shot Learning is a popular approach that aims to answer the above question. ZSL primarily aims to leverage auxiliary information available on seen and unseen classes to bridge the information gap caused by the data imbalance. In the context of image classification, ZSL aims to train a model on (seen) classes and aims to perform well on (unseen) classes not seen during the training. It can be seen as learning a mapping between the auxiliary information and classes, and use this mapping to recognize the unseen classes.

Generalized Zero-Shot Learning. The primary problem of ZSL is its practicality; training on a set of classes while aiming to classify a disjoint set of classes is not realistic. A more realistic setting is where we train on a set of classes, and aim to perform well on the training classes as well as a disjoint set of unseen classes. This setting is called Generalized Zero-Shot Learning (GZSL), where the setting more closely imitates human visual system compared to ZSL [7]. Figure 2.4 includes a diagram showing the differences between ZSL and GZSL settings in training and testing stages.

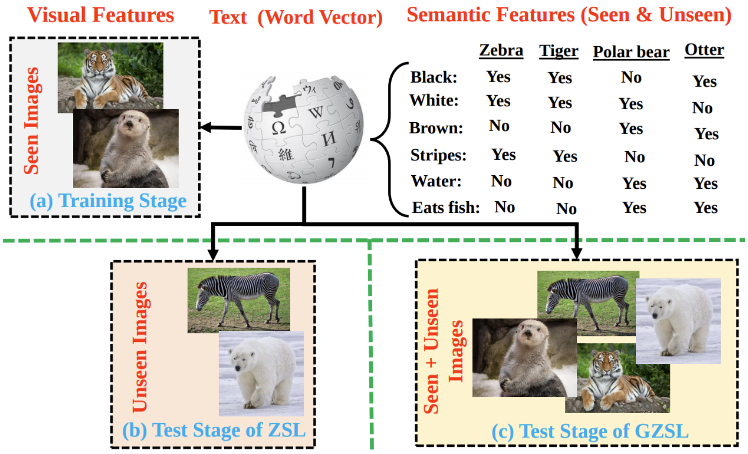
Formally, assuming a training set $S = \{(x_n, y_n), n = 1 \cdots N\}$ with $y_n \in Y^{tr}$ where Y^{tr} correspond to the training classes, the aim of ZSL is to learn a function $f(\cdot; W)$ that minimizes the regularized empirical risk [59]

$$\frac{1}{N}\sum_{n=1}^{N} L(y_n, f(x_n; W)) + R(W) \tag{6}$$

where $L(\cdot)$ is the loss function and $R(\cdot)$ is the regularization term. More specifically, we aim to learn $f(\cdot; W)$ which can be defined as

$$f(x\cdot; W) = \underset{y \in Y}{\operatorname{argmax}}\, F(x, y; W) \tag{7}$$

The primary difference between ZSL and GZSL settings is that during ZSL evaluation, $f(\cdot; W)$ is expected to assign to test images a label $y \subset Y^{ts}$ whereas in GZSL, it is expected to assign a label $y \subset Y^{tr+ts}$, where Y^{ts} refer to unseen image labels.



	Zebra	Tiger	Polar bear	Otter
Black:	Yes	Yes	No	Yes
White:	Yes	Yes	Yes	No
Brown:	No	No	Yes	Yes
Stripes:	Yes	Yes	No	No
Water:	No	No	Yes	Yes
Eats fish:	No	No	Yes	Yes

Figure 2.4 ZSL and GZSL settings. Image credit: [7].

Evaluation metrics. Throughout this thesis, we focus on the image recognition task within ZSL/GZSL settings. We follow and report the metrics widely used in the literature; normalized (i.e. averaged across all classes to treat all classes equally, regardless of per-sample count) top-1 accuracy on seen (Acc_s) and top-1 accuracy on unseen (Acc_u) classes are used. Additionally, to present a single metric, harmonic score (H-Score) is also reported, which is defined as

$$H = 2 * \frac{Acc_s * Acc_u}{Acc_s + Acc_u} \tag{8}$$

H-Score is especially useful to see the imbalances in unseen and seen class performance, and due to the inherent low performance on unseen classes, generally favour unseen classes more than seen counterparts.

Auxiliary Information. A crucial component of ZSL methods is the availability and the nature of the auxiliary information. This auxiliary information is the only bridge linking seen and unseen classes, therefore their accuracy and informativeness are of paramount importance. Note that auxiliary information is also referred to as side information, class prototypes, semantic information or class embeddings as well. In principle, any kind of information can be used as auxiliary information (i.e. discrete, continuous, etc.), but the two types often used in the literature are i) attributes and ii) word vectors [60].

Attributes are sets of information about classes provided by human annotators, such as their shape, color or even more abstract characteristics (i.e. season in the image, whether a bird has a beak or not, etc). Despite being highly accurate, it is hard to scale attributes to large datasets. However, they are the most commonly used type of auxiliary information in the ZSL literature. Word vectors are essentially automated attributes extracted from large text corpora, where classes are represented with vectors that represent various characteristics. In contrast with manual attributes, they are easier to generate and scale, but tend to have more noise. Note that we use manual attributes throughout our thesis, specifically the ones provided by publicly available datasets [59, 61, 62].

Challenges. ZSL and GZSL have their unique challenges. The first challenge is the *knowledge transfer* from seen to unseen classes, which is at the very core of both settings. An accurate mapping between visual and attribute (i.e. semantic) space must be learned, such that it facilitates the required knowledge transfer. Another core challenge is the *projection domain shift*. This is essentially domain shift on steroids; unlike the domain shift problem where two domains are known to have the same classes, in ZSL/GZSL two domains might not share the same classes. Essentially, seen and unseen classes have different classes, and they are likely to be from different domains as well. The third challenge is the *auxiliary information generation*; they are integral to the success of ZSL/GZSL methods and they must be generated in scale, with high effectiveness [7]. Last but not least, another challenge is the *overfitting*. Apparent in GZSL, many methods inherently overfit to seen classes because that is the only data they see during training [63]. Appropriate regularization and successful knowledge transfer is imperative for tackling this issue.

2.2.2. Existing Methods

ZSL/GZSL literature is quite vast, and it is an ever expanding field due to its low sample complexity nature. Earlier ZSL methods adopted a two-stage approach, where attributes

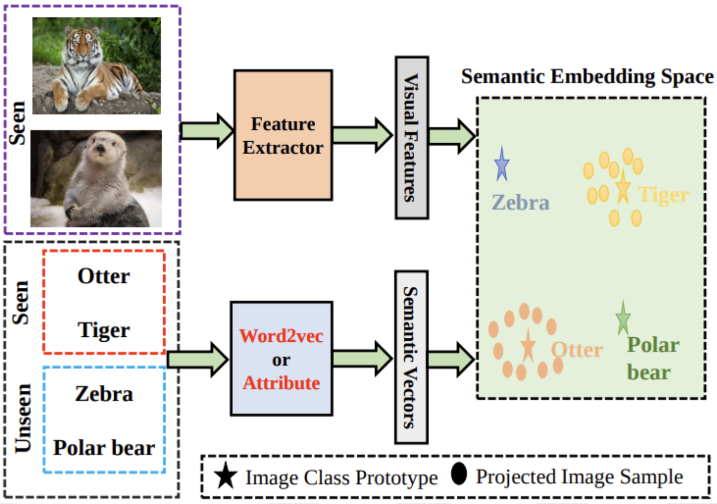


Figure 2.5 Diagram of discriminative ZSL methods. Image credit: [7].

of a given image are predicted, and then these attributes are used to find the class with similar attributes to produce the final result [64, 65]. Following these methods are modern methods which we largely divide into two; i) *discriminative* and ii) *generative* methods. *Discriminative* methods, also called embedding-based methods, learn a function that associates the visual and the semantic embeddings, which is then used to compute a similarity score between semantic vectors of unseen classes and predicted embeddings of the query image [1, 66–75]. Generative methods leverage the advances in generative modeling, and generate samples or features of seen classes based on the relation between seen classes' semantic and visual embeddings. Such methods effectively reduce ZSL into a fully-supervised problem, where the generated features or samples of unseen classes are used in training in a supervised fashion [76–79]. From another perspective, ZSL/GZSL methods are divided as *inductive* and *transductive*, where inductive methods are accepted as the conventional ZSL setting where only the visual features of seen classes and semantic embeddings of all classes are available. However, in the transductive setting, unlabeled samples of the unseen classes are also used [1, 80, 81]. The transductive setting is a mixed bag; inclusion of unlabeled samples of unseen classes defeats the very purpose of ZSL/GZSL methods, but it is not unrealistic to think that this will be possible in real-life setting [7].

Discriminative methods have various approaches in itself, where the projection (i.e. compatibility) function can be learned from visual to semantic space [66, 82, 83], semantic to visual space [84, 85] or both of them can be mapped to a shared latent space [86, 87]. The way this function is implemented varies dramatically; graphs [67, 68], meta-learning [69, 70], attention [1, 71], compositional learning [72], various network architectures [73] and even out-of-distribution detection methods are some examples [66]. In this thesis, we specifically focus on two methods; the seminal Attribute-Label Embedding (ALE) method [83] and Latent Feature Guided Attribute Attention (LFGAA) method [1]. These two methods form the backbone of our analyses presented in Chapters 3. and 4.. See relevant chapters for details on these methods. A representative diagram for discriminative ZSL methods is shown in Figure 2.5.

Generative methods primarily aim to reduce ZSL to a fully supervised problem by generating samples or features for unseen classes. In principle, they address the *projection domain shift* problem *if* they manage to generate adequate samples/features for the unseen classes. GANs [88–91] and VAEs [92–94] are the most commonly used generative methods, and they come with their burdens; unstable training, mode collapse and lack of details in generated samples (i.e. blurriness) are some example issues plaguing the generative approaches. Note that generative methods are in a comfortable lead in ZSL/GZSL benchmarks despite the unique issues they face. In this thesis, we *do not* focus on generative models and therefore do not go into further details here (see Chapter 3. for why we do not focus on generative methods).

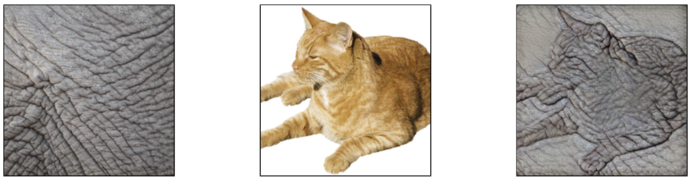
2.3. Robust Generalization

In this section, we focus on the relevant information on adversarial and corruption robustness through the lens of the more general robust generalization discussion. We motivate the need for robust generalization, specifically focus on adversarial and corruption robustness due to their relevance to our thesis and then outline the existing challenges in the field.

2.3.1. Preliminaries.

Although ML methods introduced significant advances in numerous fields, they have their unique challenges. Among all of them, quite possibly the most important one is *generalization*. The core idea behind generalization is simple; the reaction of the model to new data. A model that is said to generalize well will perform well across new data. The key point here is the *definition* of new data. Formally, when we talk about generalization in ML, we first think of how well the model performs in the test set, which it has not seen during training or hyperparameter tuning phase. Note that the training and test set of a dataset is considered to be in the same distribution. Let us call this level of generalization as the simple *train-test* generalization. Going one step further, let us assume multiple datasets with same categories; i.e. two separate cat/dog classification datasets A and B. Ideally, we want a model we trained on A or B perform well on the other. This is often the most practical case of generalization; we would like to learn robust representations of classes that will perform well *generally* across different distributions of data. Let us call this *cross-dataset generalization*. Note that this distribution change in cross-dataset generalization is essentially a *domain shift* issue that domain adaptation tries to solve.

Often what comes to mind when talk about *cross-dataset generalization* is seasonal changes, viewpoint changes, color changes, etc. However, the real-life generalization demands much more than that. Assume a query image belonging to a category seen in the training, the misclassification of the query image is *always* a generalization issue. The underlying reason of such errors might be *anything*, distribution changes, underfitting, any transformation that does not change the label, etc. Note that the issue can stem from the training data, as well as the architecture of the model. It is shown that ImageNet pretrained models often fit to the texture rather than shape [8] (see Figure 2.6), and this problem can be alleviated to a degree with injecting shape bias to the data [8]. There are datasets where extremely *hard*, yet *natural* examples are curated, where it is hard for even us humans to correctly predict the class of the image [95]. The inability of ML models to generalize across from natural images to sketches/renditions [33], which us humans can effortlessly do, is considered another example



(a) Texture image
81.4% **Indian elephant**
10.3% indri
8.2% black swan

(b) Content image
71.1% **tabby cat**
17.3% grey fox
3.3% Siamese cat

(c) Texture-shape cue conflict
63.9% **Indian elephant**
26.4% indri
9.6% black swan

Figure 2.6 ImageNet pretrained models focus on texture. Image credit: [8].

for data-based issues as well. From an architectural point of view, it is shown that CNNs tend to leverage high-frequency information imperceptible to us humans, which leads to test errors that degrades generalization [35, 39].

Among this sea of potential reasons, we focus on the specific ones that are the driving force behind the *robust generalization* discussion, which we define as the ability to perform well on a large distribution of images, including the *adversarial* and *corrupted* versions of the test images[1]. These adversarial and corrupted version of images often look the same to us humans, or we can still tell the actual class of the image, but they can completely invalidate even the most accurate of ML models. They have the same practical effect on ML models and their adoption in real-world use cases, but have *wildly* different underlying characteristics. The motivation our their selection, however, is the same; they have *immediate* and *profound* effects on generalization, and any analyses on and solution to them are of paramount importance.

2.3.2. Adversarial Robustness

What is adversarial robustness? Adversarial robustness focuses on model generalization; specifically the ability of a model to correctly classify *clean*, as well as *adversarially perturbed* images. Such images are often called *adversarial examples*, and they are first highlighted in [30]. Adversarial images are often imperceptible and created via adding learned perturbation matrices to query images, and primarily have two properties; i) humans

[1] Note that robust generalization is essentially a subset of the broader *true generalization* discussion.

can not distinguish clean and corrupted versions and ii) they lead to high-confidence, often model-agnostic misclassifications by the model. There are various theories as to why they exist; the linearity hypothesis [96], excessive dependency on high-frequency content [39], decision boundary flatness [97] and inherent uncertainty on model predictions [98] are some theories that attempt to justify their existence. In practice, adversarial examples are created with *adversarial attack* methods, and are addressed with *adversarial defense* methods.

> *'Fool me once, shame on you.'*
>
> *Anthony Weldon*

Adversarial attacks. The literature on adversarial attacks is extremely vast and virtually impossible to cover in detail in a short section. Following the seminal work that first introduced adversarial examples [30], the first attack that scaled to ImageNet was reported in [99]. A plethora of works followed; one-step attacks [96], transformation attacks [100], one-pixel attacks [101], attacks not constrained by any ℓ-norm (i.e. potentially perceptible) [102], iterative attacks [103], Jacobian-based attacks [104], universal attacks that worked for an entire dataset [105] and generative method-based attacks [106] are some examples. These attacks have various so called *threat models*, where the attacker makes an assumption about their knowledge on the system to be attacked; white-box attacks assume extensive access to the target model, its architecture and gradients whereas black-box methods can only query the target model and have absolutely no knowledge about the model [2] [108]. Attacks are also divided into two categories as *targeted* or *non-targeted*, depending on whether the attack design dictates a specific class for misprediction. In other words, targeted attacks forces a model misclassify an image as a cat, whereas non-targeted attacks just aims for misclassification where any mispredicted label would suffice. *Transferable* term is coined to indicate whether an adversarial example can invalidate a different model than the one it is optimized to fool. Adversarial attacks have also managed to make their way into the real-world from the digital world *much like Agent Smith*, where physical attacks, even when

[2] Note that there is a third variant *grey-box attacks* [107] that is not as applicable as the other two, and not quite relevant to our thesis. Therefore, we omit its discussion here for brevity.

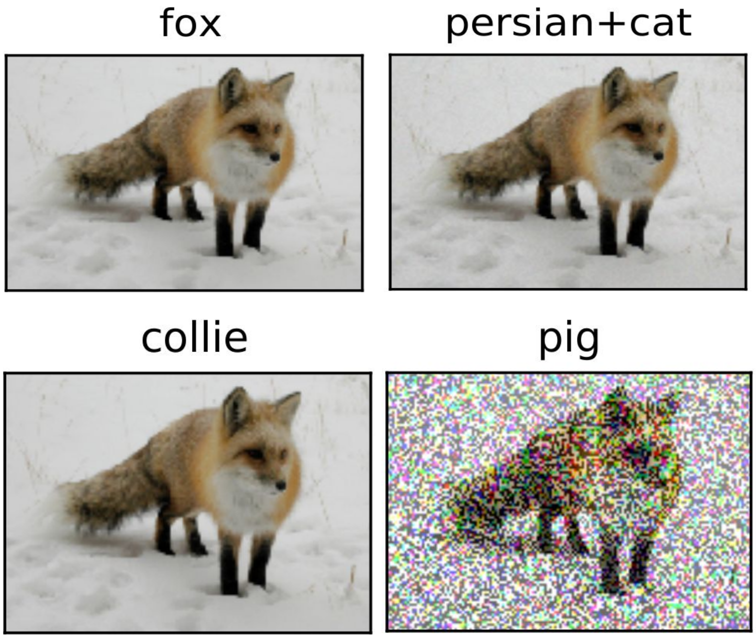


Figure 2.7 Original (top left) and adversarial versions of a fox image.

printed on a t-shirt or on a sticker, managed to fool real-life ML systems [109]. Naturally, adversarial attacks are not limited to just image recognition, but extended to various other vision tasks like object detection [110], depth estimation [111] and object tracking [112], as well as other modalities such as NLP [32], speech [31], point clouds [113], visual question answering [114] and many more. See Figure 2.7 for visual examples of adversaries with different strengths.

> *'Fool me twice, shame on me.'*
>
> *Anthony Weldon*

Adversarial defenses. Similar to adversarial example literature, adversarial defense literature is quite vast as well, therefore we only highlight the important advances. Adversarial defenses can be divided into three; methods that i) change the training regime, ii) change the network and iii) use additional modules during inference [108]. A prime example to the first category is the popular adversarial training, where adversarial examples are added to the training sample pool [30, 96]. It is succeeded by improved methods

[115–117], yet it is still considered as a strong baseline. Input gradient regularization [118], autoencoder-based purification [119], bounded ReLUs [120], feature regeneration [121] and knowledge distillation [122] are some methods for the second category. The third category, and the second category to some degree, can be further divided into two sub-categories where adversarial attacks are either purified or simply detected. Total variance minimization [123], bit-depth reduction [124], JPEG compression [125], GANs [126] or other network architectures [105] have been used for adversary purification, whereas feature squeezing [124] and specialized detector architectures [127] have found use for adversary detection. Similar to adversarial examples, there are many defense mechanisms designed for vision tasks other than image recognition [128, 129], as well as other data modalities such as speech [130], NLP [131] and many more. Among all the attacks and defenses listed, we have used several of them in our research and will go into more detail in the next chapters. Note that adversarial attack and defense literature is experiencing an ongoing arms race, with no end in sight.

2.3.3. Corruption Robustness

From adversaries to corruptions. Despite their importance, adversarial examples require several key ingredients; crafting an adversarial example with imperceptible changes require expertise as they are arguably the *worst-case* scenarios for the model. This often means that there needs to be *bad intent* for such examples to be realized; it is not quite realistic for a system and/or a third party to create an adversarial example purely by chance and/or without bad intent. These facts do not mean adversaries are not important, but rather makes us question the threat they pose.

Common image corruptions are more *practical* scenarios where we see them happen naturally; i.e. weather effects, digital artefacts, sensor noise, etc. They are not necessarily the *worst-case* (i.e. imperceptible) or require bad intent to materialize, but they *do occur* and occur *more frequently*. The elevated frequency of occurrence for corruptions make them a more severe threat to real-world ML systems, and also a more practical venue for robustness analyses. As one expects, *corruption robustness* focuses on the ability of a model to correctly classify *clean* as well as *corrupted* images. Note that corruptions, especially due to their unbounded nature, are arguably even more dangerous than adversaries and are shown to easily invalidate otherwise state-of-the-art models [9].

The corruption robustness field is comparably younger than the adversarial robustness field, and thus still evolving. Its consolidation and standardization begun with the seminal work that introduced ImageNet-C [9] dataset. ImageNet-C is essentially the ImageNet validation set, but with several image corruptions with some 15 different types and 5 severity levels, such as noise types, weather effects, digital artefacts and commonly occurring blur types. These effects are simulated synthetically and act as a proxy to naturally occurring corruptions, where models are evaluated on these images to assess their performance against such corruptions. Practitioners are encouraged *not to train their models using images corrupted* with the same categories (i.e. unlike adversarial training where adversaries are used in the training pool). Also, an additional four types of corruption representing each category are provided as *validation* corruptions, where practitioners are encouraged to validate their models on these four types, rather than the test corruptions directly. ImageNet-C led to the release of multiple relevant datasets, such as CIFAR-C and MNIST-C [9, 132] with relevant corruption types. ImageNet-P is released to assess *perturbation robustness*, to see if models can preserve their predictions under temporally evolving, potentially cascaded corruptions over time. The types of corruptions specific to other modalities [133, 134] or more geometrically grounded corruptions [135] have been standardized, and vision tasks other than image classification [136, 137] have been evaluated against corruptions. See Figure 2.8 for example corruption visualizations.

Improving corruption robustness. Similar to the arms race between adversarial attacks and examples, there has been a plethora of papers improving the corruption robustness of existing models. These methods primarily focus on devising data augmentations to diversify the training distribution such that the models become much more robust to corruptions. Cascading randomly sampled image augmentations [36, 138], new augmentation regimes [139–142], leveraging unlabeled data for self-supervised pretraining [143], adversarial

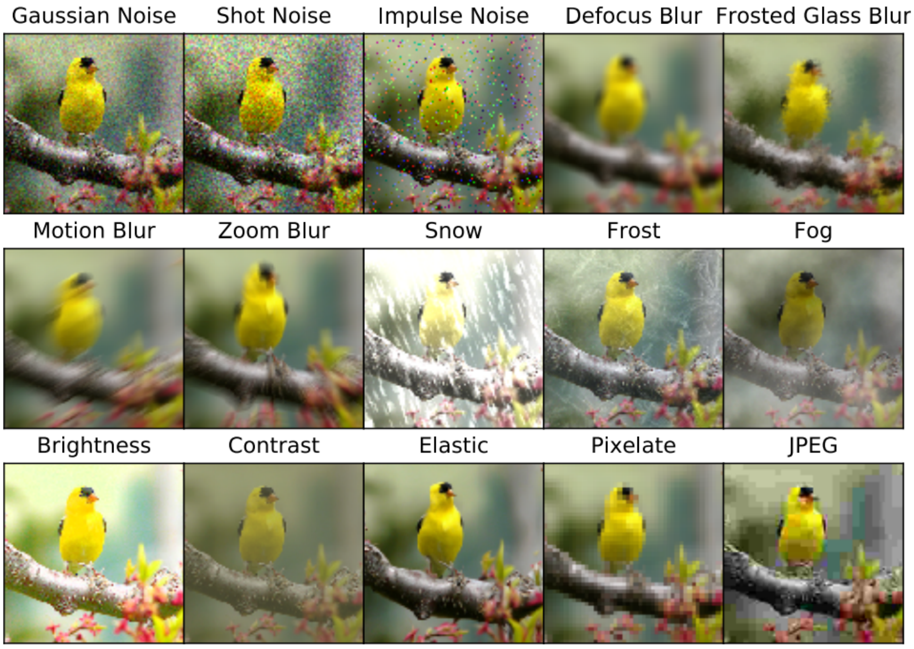


Figure 2.8 ImageNet-C corruptions. Image credit: [9]

training [144, 145], explicit shape-bias injection [146, 147] and style-transfer [148] are examples to such augmentations. Several methods focused on batch normalization and its effect on corruption robustness, which turned out to be decisive factor in overall model robustness [149]. This led to intra-batch sample statistic augmentations that aimed to focus models' attention to what matters [150]. Model ensembles, formed out of different domain expert models [151] or frequency-biased models [152] are shown to be helpful as well. Enhancing sub-networks of an architecture [153], augmentation consistency losses [154] and wavelet-transform enhanced architectural changes [155] also improve robustness. Last but not least, based on the frequency-spectra analyses of corruption robustness, several methods have been proposed; biasing models to low-frequency regions [156], amplitude-phase mixing for data augmentation [10] and spectrum perturbation [157, 158] are shown to be effective for improving robustness. Note that many of these augmentation methods, unlike adversarial defenses, aim to address the broader robustness issue, where they aim to improve both clean, adversarial and corruption accuracies, with varying degrees of success. Many of these methods are either borrowed/adapted from existing augmentation literature, or went on to be an integral part of augmentation literature and not just robustness literature. Note that among all these methods, we have used several of them in our research and will go into more detail in the next chapters.

2.3.4. Challenges

Despite the stunning advances in robust generalization, there are still numerous challenges that needs to be addressed. A primary challenge is the very *definition of corruption*; existing methods either simulate a subset of existing corruptions or focus on an extremely small subset of (i.e. deblurring datasets) real-life corruptions. As is the case for any progressing field, existing methods are likely to reach the limits of existing datasets at some point in the near future, which will require new benchmark datasets with significant scale and diversity. Another challenge is *the robustness-accuracy trade-off*; it is often discussed that these two are in odds at each other, where increased robustness leads to worse performance on clean images, and vice versa [159]. Achieving a more substantial impact would require the improvement on both fronts, not just one. The final challenge (among an extensive list of others not mentioned here) is *the severe bias of existing studies to fully supervised methods*; existing robustness methods exclusively focus on fully supervised settings which limits the discussion to a restricted set of use cases. Effects of low model accuracy, data imbalance, imperfect supervision profiles and the resulting complex decision boundaries are among many areas relatively unexplored in the literature. Our thesis is motivated by this challenge specifically, hence our focus on assessing zero-shot learning robustness on multiple fronts.

2.4. Frequency Spectra

In this section, we focus on the relevant information of frequency image analyses. We first motivate why frequency analyses in images are necessary, when and where they are useful and their relevance to our thesis.

2.4.1. Preliminaries

Images inherently have a *spatial* form, where our perception is dependant on its width and height. In their canonical domain, let us call it *the pixel domain*, images are represented with various color spaces; i.e. as a function of RGB intensities, hue/saturation/lightness, etc. for each pixel location. Everything is built on the smallest building block *pixels*. The pixel domain representation is integral to our perception and for many uses in computer vision and image processing, but a well-motivated question is can we have *different building blocks to represent images* that will also be useful?

We first take a step back and go back to basics. In signal processing, where signals are often one-dimensional [3] (i.e. electromagnetic waves, acoustic waves, speech, etc.), signals are often defined with their magnitude, but more importantly, their periodicity. Their *wavelength* is of paramount importance; it effects how far a signal can travel due to frequency-based attenuation levels, how much information it can provide, etc. Note that the entire electronics industry is built on these principles; in telecommunications sector, companies pay billions to governments to acquire the rights of certain frequency bandwidths in which they can operate. Therefore, it is imperative to refresh the basic knowledge of signal periodicity.

2.4.2. Primer on Fourier Transform

Jean-Baptiste Joseph Fourier, a mathematician and physicist, claimed in early 1800s that any function can be decomposed into sine waves of different frequencies. Essentially, instead of having time or space as the primary building block of a signal, this decomposition uses sine waves with different frequencies as its basis. This process, called *Fourier Transform*, effectively rewrites a signal as a weighted sum of sine waves with different frequencies; i.e. showing us how *much* of each frequency is present in the signal. Let $x(t)$ be a continuous signal as a function of time t, the Fourier Transform is defined as [160]

[3] Excluding the time dimension.

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt \tag{9}$$

where $X(\omega)$ is the original signal, now as a function of the frequency ω. The Inverse Fourier Transform reverses this process and is defined as

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{j\omega t}d\omega \tag{10}$$

Note that the very building block of this frequency domain is a sine wave $A(\sin\omega+\theta)$, where A is the magnitude and θ is the phase component. The magnitude A tells us *how much* of the that specific frequency component (i.e. amplitude) and phase tells us about its shift in time. As useful as they are, these equations are defined for uni-variate continuous functions, whereas images are multi-variate discrete functions.

Discrete Fourier Transform (DFT) comes to the rescue for digital signals that form the backbone of our lives. Digital signals sample $x(t)$ and turn it into a discrete set of observations. Assuming we have an image $x(a, b)$ with size $N \times N$, and by simply extending the uni-variate continuous Fourier transform to multi-variate discrete case, DFT is defined as

$$X(k,l) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} x(i,j)e^{-i2\pi\left(\frac{ki}{N}+\frac{lj}{N}\right)} \tag{11}$$

where i, j index the image in spatial domain, the exponential terms correspond to basis sine functions and the Fourier space is indexed by k, l. Note that the number of frequencies are the same as the number of pixels; the spatial and Fourier domain images are of the same size. Similar to the continuous case, the inverse DFT is calculated as

$$x(a,b) = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} X(i,j)e^{i2\pi\left(\frac{ka}{N}+\frac{lb}{N}\right)} \tag{12}$$

2.4.3. Frequency Analyses on Images

A good point to start is to ask this question; *what is frequency in an image*? High-frequency in images correspond to a high rate of intensity change; i.e. edges, object boundaries whereas low-frequency in images correspond to no or gradual change areas. Phase and amplitude components are stored for each frequency, but unlike the 1D continuous signal case, they are a bit harder to interpret for 2D discrete images. Magnitude is still straightforward; it tells us how much signal there is at a specific frequency component. Things get a bit messy when it comes to phase; shown in Figure 2.9 are an input image, and its phase and (zero-centred) amplitude [4]. Unsurprisingly, neither of them mean something to us at first glance, especially the phase. An interesting experiment is shown in the bottom of Figure 2.9; once the phase and amplitude of two images are combined, and inverse DFT is applied, the resulting image looks more like the image we took the phase component from. Therefore, we can say that the phase encodes important spatial information regarding where each frequency component occurs in an image.

How to use frequency analyses in practice? Image frequency analysis is used in several crucial applications. Many compression algorithms (including the famous JPEG [161]) filter out the high-frequency components of the image, which commonly occurring noises reside. Upsampling and downsampling operations are carefully designed to prevent *aliasing*, a phenomena directly related to frequency bands of an image. Edge detection algorithms often exploit frequency information of an image, which are used in more abstract tasks like segmentation as well. From a computational perspective, using Fast Fourier Transform, convolution operations can be calculated quite efficiently in the frequency domain, leading to performant applications. Finally, unlike the original pixel domain representation that focuses on the *where*, the frequency domain representation of images focus on the *what*. Each k, l in $X(k, l)$ (Equation 2.4.2.) correspond to information about *all pixels* in the image $x(a, b)$. This leads to many interesting applications, such as neural network designs leveraging this fact to enlarge their receptive fields [162]. Our thesis heavily leverages frequency analyses

[4] We use this interchangeably with magnitude.

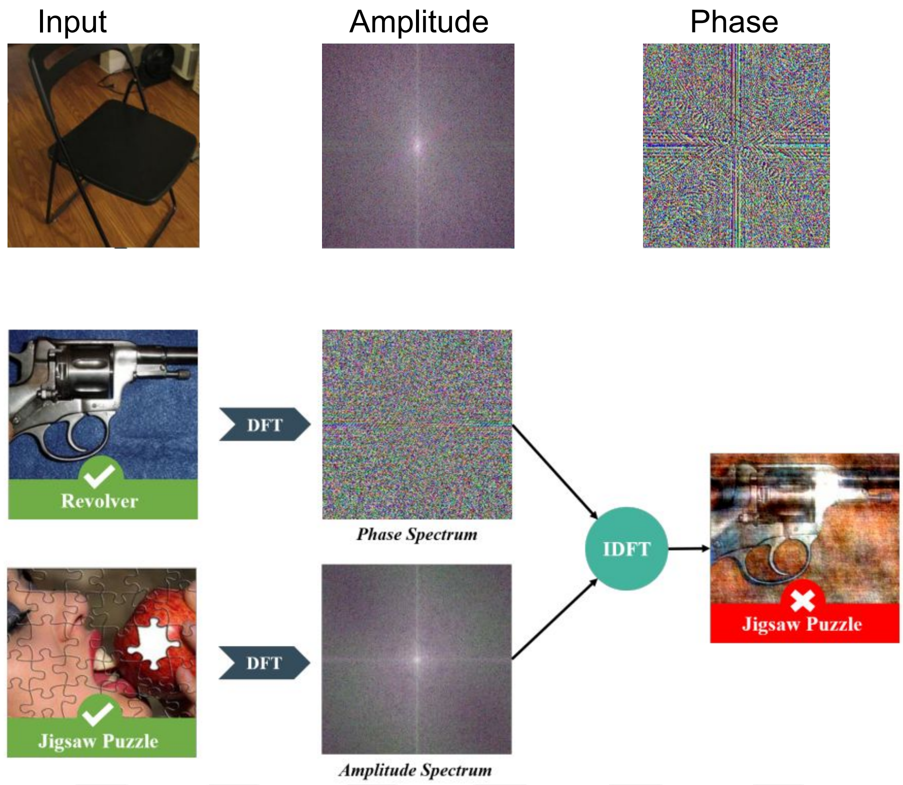


Figure 2.9 Mixing the phase and amplitude of two images. Image credit: [10]

in Chapter 5., where we use the fundamental frequency analyses theory to devise a data augmentation method to improve model robustness.

3. A Deep Dive into Adversarial Robustness in Discriminative Zero-Shot Learning

In this chapter, we present our work on adversarial robustness in discriminative ZSL models. We show that such ZSL models behave somewhat different to fully supervised models when adversaries are present, and we lay out the reasons with detailed analyses; the effect of dataset, model maturity, class boundary transitions induced by adversaries, effects of defenses and the discrepancy between ZSL and GZSL performance. We also identify and discuss the *pseudo-robustness effect*.

> *'As soon as you start fulfilling your purpose your adversaries will appear.'*
>
> *Sunday Adelaja*

3.1. Introduction

The swift ascension of machine learning models ignited a new wave of state-of-the-art (SOTA) results in various fields; computer vision, natural language processing (NLP) and speech recognition are just few examples. The increase in data availability, compute budgets as well as core advances, led ML models to a meteoric rise in various tasks with no apparent signs of a slowdown.

However, authors of [30] showed that ML models suffer from adversarial examples, which are carefully crafted noise patterns that can lead models into mispredictions. These noise patterns misguide models while introducing imperceptible perturbations to the given sample. Starting with vision, adversarial intrusions have been extended various modalities such as speech [31], NLP [32] and many others [163]. An equal attention has been given to defend the models against such attacks, either by robustification of models or introducing mechanisms to detect and/or purify adversaries [122]. Adversarial ML first emerged by focusing on *toy* datasets in vision, such as MNIST and CIFAR-10/100, but swiftly scaled their threat to larger benchmarks such as ImageNet, and even to commercial solutions [164, 165].

A large portion of the adversarial ML literature focused on supervised solutions and targeted to improve their robustness. In Zero-shot learning (ZSL) and Generalized Zero-shot learning (GZSL), however, the goal is different from the well-known supervised approach; the aim is to learn from a subset of classes, such that we can optimize the transfer of the learned knowledge to a set of classes which are completely unseen during the training. The inherently tough problem of ZSL is still an open research topic, despite the recent advances. The introduction of adversarial examples to ZSL models is an interesting intersection due to three primary reasons. First, ZSL and adversarial robustness essentially try to address the same thing, which is out-of-distribution sample recognition. In ZSL, these samples are from different classes where adversaries are the misclassified, same-class samples. Second, the complex knowledge transfer from seen to unseen classes is already hard, and the introduction of adversaries will only make things more challenging. Third, discriminative ZSL models are prone to attacks from multiple mediums; the attacks can be made on images, thus visual embeddings, or on the attributes, thus the semantic embeddings. All these reasons make the problem more appealing as their analyses can shed light on the broader generalization issue.

In this chapter, we present a thorough set of analyses on ZSL techniques and their reaction to adversarial intrusions [5]. In contrast with the recently popularized approaches where ZSL is reduced to a supervised problem [78, 79], we look back in time and fix our gaze on the label embedding model [83, 166] and assess its reaction against widely used adversarial attacks and defenses. Through thorough assessment on popular benchmarks, we present a framework that analyses the algorithm, but also presents rigorous analyses on many other aspects; the effect of datasets (i.e. per-class sample count), the trends in class boundary transitions as well as the details on how adversaries affect the knowledge transfer from seen to unseen classes. We hope that this work will be the start a discussion on the adversarial robustness of ZSL models, which has surprisingly, and largely, been ignored. Furthermore,

[5] Code is available at `https://github.com/MKYucel/adversarial_robustness_zsl`

we believe that this chapter can serve as a benchmark for future work, and guide researchers towards improving ZSL model robustness as a whole.

3.2. Related Work

Adversarial attacks. Adversarial ML has been an important segment of machine learning in previous years as it shed light on important drawbacks of widely used models. It has been shown in [30] that a noise pattern can be generated by aiming for the misprediction of a given sample image; within certain ℓ-norm constraints, the generated noise pattern might be completely imperceptible to us humans. A popular alternative is a one-step, fast attack that leverages the gradients of the loss function with respect to the parameters of the network, with the aim of generating a perturbation [96]. Another approach approximating the possible minimum perturbation required for a misprediction is presented in [103]. A significantly revised version of [30] is proposed in [99], where not one but three variants of the original attack, each leveraging a different ℓ-norm to restriction to generate the perturbations, are shown to scale to ImageNet and by-pass existing defense mechanisms [122].

The literature on the adversarial attacks is quite large; ranging from attacks universal to a dataset [105] to attacks changing only a single pixel [101], spatial transformation attacks [100] to attacks focusing on most important pixels [104], attacks not constrained with any ℓ-norm [102] to black-box attacks[165], and even physical attacks [167] are some examples in the literature. Adversarial intrusion exist practically anywhere ML is used, such as reinforcement learning [168], NLP [32], graph networks [169], LIDAR [163], other vision tasks [110, 170–172], speech [31] and even commercial solutions available to customers [173].

Adversarial defenses. Numerous defenses aimed to combat adversaries have been presented, leading to a cat-mouse game between the two. Several widely known defense methods are knowledge distillation [122], adversarial training [30], label smoothing [116], input-gradient regularization [118], ReLu activation analyses [120], feature generation [121], generalizable defenses [174], GANs [126] and auto-encoders [127] for the purification of adversaries and metric learning [175]. These methods either re-train the model or propose an additional module that identifies or neutralizes the adversaries. Another line of defenses have leveraged existing advances and leveraged them against adversaries; JPEG compression [125], bit-depth reduction, spatial smoothing [124] and total variance minimization [123] are some examples. Adversarial ML literature is quite large; refer to [108, 176] for thorough reviews on the topic.

Zero-shot learning. A large part of ML tasks have acquired SOTA results using supervised settings, where all classes have ground-truth that drives the learning process. However, the collection and curation of a fully-labeled dataset leads to a bottleneck in scaling up ML models in practice. The laborious requirement of ground-truth label collection can be partially addressed by the (transductive) self-supervised learning methods [177], but the unlabeled data or the auxiliary supervision are not guaranteed to be available. Zero-shot learning aims to tackle this by effectively closing the gap between *seen* (i.e. classes available during training) and *unseen* (i.e. classes unavailable during training) classes by transferring the knowledge learned during training. Generalized Zero-Shot learning, on the other hand, performs this knowledge transfer while keeping the accuracy values on the *seen* classes high as well. The side information on seen and unseen classes is leveraged to close the gap.

ZSL approaches, in its infancy, had a two-phase setting, where the attributes of an image were predicted and these attributes were used to find the class with the most similar attributes [64, 65]. By directly learning a linear [82, 83, 178, 179] or a non-linear compatibility [66, 180–182] function to map from visual to a semantic space, later models changed to a single-phase setting. The reverse mapping, from semantic space to the visual space [84, 85], has also been explored. Embedding visual and semantic embeddings into a common latent space for ZSL have also proven to be useful [86, 87]. Transductive approaches leveraging visual or semantic information on unlabeled unseen class samples [80, 81] are another discriminative approaches. Recently, in addition to discriminative approaches [74, 183], generative approaches [76, 78, 79, 184] which model the mapping between visual and semantic spaces are increasingly being used to generate samples for unseen classes, eventually reducing ZSL to a supervised setting. For further information on ZSL and GZSL, readers are referred to [55, 185].

A recent unpublished paper [186] proposed a ZSL model robust to adversarial attacks by formulating an adversarial training regime. Our study, on the other hand, concentrates on setting up a framework and presents a benchmark to guide researchers' efforts towards adversarially robust ZSL/GZSL models, by presenting analyses of existing datasets and the effects of several attacks and defenses. Note that our study is the first to establish such a benchmark.

3.3. Model Selection

We take a step back and focus on the models that aim to transfer the knowledge learned from seen classes to unseen classes, unlike the recent ones that rely on generative models to generate samples using class embeddings for unseen classes and try to reduce ZSL to a supervised setting. We believe that focusing on the latter would mean evaluating the sample generation mechanism for adversarial robustness, rather than evaluating the robustness of the model that aims to realize seen/unseen class knowledge transfer.

As presented in Section 3.2., there are numerous suitable candidates for model selection. We select the label-embedding model [83], which has been shown to be a stable and competitive model even in modern benchmarks [185]. Attribute-label embedding (ALE) model is formulated as

$$F(x, y; W) = \theta(x) W^T \phi(y) \tag{13}$$

where $\theta(x)$ is the visual and $\phi(y)$ is the class embeddings. These two modalities are associated through the compatibility function $F(\cdot; W)$.

We select ALE due to the fact that it is one of the first studies that showed direct mapping by leveraging data and auxiliary information is more effective than the intermediate attribute prediction stage. Although there are methods which build on what ALE does [6], we believe the results of ALE will be representative of the adversarial robustness of this family of ZSL methods. Individual analyses of more approaches are encouraged, but are not in our current scope. We note that we focus on an inductive setting.

3.4. Attacks and Defenses

Threat Model. We choose three white-box, per-image attacks where the attacker has full access to the model architecture and its parameters. The attack model assumes a setting where *all* images are attacked, regardless of their original predictions (i.e. whether they were classified correctly or not). We choose a training-time defense (i.e. robustifying the model by re-training) and two data-independent, preprocessing defenses, where input images are processed before being fed to the network. The defense model assumes a *blind* regime, where none of the defenders have access to attack details nor the attack frequency (i.e. defenses are applied to all images; regardless whether the attacks introduced misclassifications or not). In the next sections, we present the chosen attacks and defenses.

Attacks. *The first attack* is the popular Fast Gradient Sign method (FGSM) attack [96] that is based on the *linearity hypothesis*. By taking the gradient of the loss function with respect to the input, the change of the output with respect to each input component is estimated. This is used to craft adversarial perturbations that will guide the image towards these directions, which means *maximizing* the loss with respect to input components. We select FGSM due to its one-shot nature (i.e. no optimization), its low computational complexity and the fact that it is *not* optimized for the minimum perturbation. FGSM is formulated as

$$\eta = \epsilon sign(\nabla J(\theta, x, y)) \tag{14}$$

where $\nabla J()$ is the gradient of the cost function J with respect to the input image. We use the *untargeted* version of FGSM that restricts the perturbation with the ℓ_∞-norm.

[6] As noted in [185], models focusing on linear compatibility functions have the same formulation, but different optimization objectives.

The second attack is the *DeepFool* [103] attack. DeepFool concentrates on the distance of an image to the closest decision boundary. It computes the distance to some number of decision boundaries, finds the closest and takes a step towards this boundary. For non-linear classifiers, this is approximated by an iterative approach that tries to cross the boundary, until an iteration limit or the boundary is crossed. We select DeepFool due to following reasons; i) it is an optimization based attack, ii) it aims for the minimum perturbation iii) it is inherently indicative of the decision boundary characteristics. We use the original *untargeted* version that restricts the perturbation with the ℓ_2 norm. DeepFool is formulated as

$$r_*(x_0) = \frac{|f_{\hat{l}(x_0)}(x_0) - f_{\hat{k}(x_0)}(x_0)|}{\left\| w_{\hat{l}(x_0)} - w_{\hat{k}(x_0)} \right\|_2^2} \left(w_{\hat{l}(x_0)} - w_{\hat{k}(x_0)} \right) \tag{15}$$

where $r_*(x_0)$ is the minimum perturbation for the closest decision boundary $\hat{l}_{(x_0)}$, w is the classifier, x_o is the query image and $\hat{k}$ is the mapping done by the classifier.

The last attack is the *Carlini-Wagner* [99] attack. It essentially refines the loss proposed in [30] through several aspects and propose three different attacks, where each attack uses a different ℓ-norm constraint to restrict the perturbation. We select it due to several reasons; i) it is one of the first attacks that beats an adversarial defense, ii) one of the first to scale to ImageNet and iii) it is a high-performing attack. We use the *untargeted*, ℓ_2-norm version for a better comparison with DeepFool. It is formulated as

$$minimize \left\| \frac{1}{2} tanh(w) + 1) - x \right\|_2^2 + c.f \left(\frac{1}{2} tanh(w) + 1 \right) \tag{16}$$

where x is the input image, the term $\frac{1}{2}(tanh(w) + 1) - x_i$ is the changed variables to meet box-constraints and c is a constant whose value is found by binary search. f is given as

$$f(x') = max \left(max\{Z(x')_i : i \neq t\} - Z(x')_t, -\kappa \right) \tag{17}$$

where t is the target class, κ is the constant controlling the confidence of misclassification and x' is desired adversarial example.

Defenses. *The first defense* is the *label smoothing*. It is a popular regularization method that avoids over-confident predictions. It assigns soft labels to ground-truth labels and is formulated as

$$p(y|x_i) \begin{cases} 1 - \epsilon + \epsilon n(y|x_i), & \text{if } a = 1 \\ \epsilon n(y|x_i), & \text{otherwise} \end{cases} \tag{18}$$

where $p(\cdot)$ is the ground-truth assignment, y_i is the correct class for input i, x_i is the input i, ϵ is the weight factor and $n(\cdot)$ is the added noise distribution. Label smoothing has been shown to be a good defense method [116, 187] and its success is explained by the prevention of confident predictions on out-of-distribution samples. We select label smoothing as it is i) a training-time defense, ii) simple yet effective and iii) it is a good use case of ZSL models.

The second defense is *local spatial smoothing*. It has been reported that feature squeezing techniques [124] provide adversarial robustness as they shrink the feature space where adversarial examples reside. We use median-filter with reflect-padding to preprocess images before feeding them to the network. We select spatial smoothing due to i) its data and attack-independent and ii) its inexpensive nature. Furthermore, testing it against non-l_0 attacks is a good use case for its efficiency [7]. We only use the spatial smoothing operation, and do not use the detection mechanism.

The last defense is the *total variance minimization* defense. It has been proposed [123] as an input transformation defense, where the goal is to alleviate perturbations by reconstructing the image. First, some pixels are selected with a Bernoulli distribution from the adversarial image. Using these pixels, the image is reconstructed by taking into account the total variation measure. Total-variance minimization is shown to be an efficient defense as it promotes the removal of small and localized perturbations. We select it due to its simple and

[7] It has been noted in [124] that this defense is more effective against l_0-norm attacks.

data/attack independent nature. It is also an ideal candidate to evaluate different attacks due to its localized perturbation removal ability. TVM is formulated as

$$min_z \left\|(1-X)\odot(z-x)\right\|_2^2 + \lambda_{TV}\cdot TV_p(z) \tag{19}$$

where z is the reconstructed image, X is the random variable indexed by pixel locations, x is the input image, $TV_p()$ is the total variation and $\odot$ is the element-wise multiplication. The total variation measure is given as

$$TV_p(z) = \sum_{k=1}^{K}\left[\sum_{i=2}^{N}\left\|z(i,:,k)-z(i-1,:,k)\right\|_2 + \sum_{j=2}^{N}\left\|z(:,j,k)-z(:,j-1,k)\right\|_2\right] \tag{20}$$

where i, j and k are the pixel locations.

3.5. Dataset and Evaluation Metrics

The evaluation is performed on three widely used ZSL/GZSL datasets; Caltech-UCSD-Birds 200-2011 (CUB) [61], Animals with Attributes 2 (AWA2) [185] and SUN [62]. CUB is a mid-sized, fine-grained dataset with 312 attributes, where 200 classes are represented with a total of 11788 images. CUB is challenging, as intra-class variance is quite tough to model due to similar appearances and low number of samples. SUN is another mid-sized, fine-grained dataset with 102 attributes. SUN is a challenging case as well, as it consists of 14340 images of 717 classes, resulting into even fewer images per class compared to CUB. AWA2 is a larger-scale dataset with 85 attributes, where 50 classes are represented with 37322 images. AWA2, due to having a higher amount of images with fewer classes, makes generalization to unseen classes harder. We leverage the splits proposed in [185] for both ZSL and GZSL settings. We use the standard per-class top-1 accuracy for ZSL evaluation. For GZSL, per-class top-1 accuracy values for seen and unseen classes are used to compute harmonic-scores.

	Zero Shot			Generalized Zero Shot								
	C	S	A	C			S			A		
Attack	Top-1			u	s	h	u	s	h	u	s	h
Original	54.5	57.4	62.0	25.6	64.6	36.7	20.5	32.3	25.1	15.3	78.8	25.7
$FGSM_1$	40.3	47.7	42.5	18.5	45.4	26.3	17.7	25.9	21.0	10.7	58.9	18.1
$FGSM_2$	18.5	16.3	14.8	10.8	11.7	11.2	8.1	9.8	8.9	3.4	10.0	5.1
$FGSM_3$	15.2	11.8	16.4	9.0	10.2	9.6	4.3	5.5	4.9	2.2	11.2	3.7
$DEFO_1$	30.9	25.6	50.6	9.1	19.1	12.3	6.4	7.2	6.8	13.3	41.2	20.1
$DEFO_2$	30.8	25.5	50.5	9.1	18.9	12.3	6.4	7.2	6.8	13.4	41.2	20.2
$DEFO_3$	22.4	17.8	41.4	7.6	11.5	9.2	6.3	6.3	6.3	13.0	30.2	18.2
$CaWa_1$	28.9	43.1	43.2	17.0	29.0	21.4	17.7	24.9	20.7	15.2	56.3	24.0
$CaWa_2$	25.9	40.9	36.9	16.4	24.4	19.6	17.7	23.9	20.3	15.2	46.6	22.9
$CaWa_3$	24.6	39.8	34.7	15.9	23.1	18.9	17.5	23.4	20.0	15.2	43.6	22.5

Table 3.1 Results when *all* images are attacked.

3.6. Implementation Details

In order to make the computational graph end-to-end differentiable, we merge ResNet-101 [4] (used to extract AWA2 [185] embeddings) feature extractor with ALE model. To reproduce the results of ALE [185], we freeze the feature extractor and train ALE for each dataset. The reproduced values of ALE are denoted as *original* in the tables; note that there are slight variations compared to the original results in [185]. We use PyTorch [188] for our experiments.

For FGSM, we sweep with a large range of ϵ values where we end up with visible perturbations. We sweep with *maximum iteration* and ϵ (added value to cross the boundary) parameter for DeepFool (DEFO) and Carlini-Wagner (CaWa, C&W) attacks, as we observe diminishing returns (i.e. not producing stronger effects despite reaching intractable compute time) for other parameters. Specifically, we use [$FGSM_{1-3}$ ϵ: 0.001, 0.01, 0.1] [$DeepFool_{1-3}$ max_iter, ϵ: (3,1e-6), (3,1e-5), (10,1e-6)] [$C\&W_{1-3}$ max_iter: 3,6,10]. We assign *0.9* to the ground-truth class in label smoothing. For spatial smoothing and total-variance minimization, we use 3x3 windows and maximum iteration of 3, respectively.

	Zero Shot			**Generalized Zero Shot**								
	C	S	A	C			S			A		
Attack	Top-1			u	s	h	u	s	h	u	s	h
Original	54.5	57.4	62.0	25.6	64.6	36.7	20.5	32.3	25.1	15.3	78.8	25.7
SpS	49.3	53.2	59.3	21.5	56.5	31.1	20.1	28.0	23.4	14.3	75.5	24.1
LbS	52.2	55.2	60.6	22.7	56.2	32.4	18.4	31.6	23.3	16.3	74.2	26.8
TVM	51.4	54.0	60.3	24.4	60.7	34.8	19.9	29.5	23.8	12.9	76.4	22.1

Table 3.2 Results where all images are defended (without any attacks).

We apply the same attack and defense parameters for every dataset for a fair comparison of dataset characteristics. Across the tables in this chapter, *C*, *S* and *A* stand for CUB, SUN and AWA2 datasets. *Top-1* is the top-1 accuracy, where *u*, *s* and *h* are unseen, seen and harmonic accuracy values.

3.7. Results

Attacks. We first present the effect of each attack on ZSL/GZSL performance. The results are shown in Table 3.1.

In the *ZSL* setting, all attacks introduce detrimental effects at various rates, across all datasets. FGSM introduces the strongest attack as one would expect, as in its most powerful configuration, it introduces perceptible perturbations. On CUB, we see C&W attack leading in low-maximum iterations, but it loses out to DeepFool in higher iterations. On SUN, C&W fails to deliver and can not scale with the increasing maximum iterations, where DeepFool manages to do a better job with an accuracy reduction 20 points higher than C&W. On AWA2, C&W does a better job than DeepFool across all settings, with up to 7 points more reduction. FGSM causes an upward accuracy spike on AWA2, despite its increasing strength. This is mainly caused by changing the originally incorrectly predicted labels to their correct labels, thereby increasing the accuracy. Lastly, DeepFool produces diminishing returns except the highest maximum iteration, across all datasets.

	Zero Shot			Generalized Zero Shot								
	C	S	A	C			S			A		
Attack	Top-1			u	s	h	u	s	h	u	s	h
Original	54.5	57.4	62.0	25.6	64.6	36.7	20.5	32.3	25.1	15.3	78.8	25.7
$FGSM_1$	47.9	51.1	54.5	20.3	53.5	29.4	19.8	26.0	22.5	12.7	70.0	21.5
$FGSM_2$	31.9	36.0	24.6	14.5	30.5	19.7	14.5	16.6	15.5	6.2	25.3	10.0
$DEFO_1$	46.4	49.0	58.0	18.8	50.1	27.3	15.9	21.0	18.1	13.5	69.3	22.6
$DEFO_3$	46.2	48.8	58.0	18.7	50.0	27.2	15.9	21.0	18.1	13.1	68.8	22.1
$CaWa_1$	48.3	52.7	58.2	21.0	55.0	30.5	20.2	27.3	23.2	14.2	73.6	23.9
$CaWa_3$	48.4	52.3	58.2	21.0	54.9	30.4	20.0	27.2	23.1	14.2	73.3	23.8

Table 3.3 Results: *All* images are attacked and defended with *spatial smoothing*.

In the *GZSL* setting, across the board reduction for all datasets are observed for all three attacks. On CUB, DeepFool performs the best (9.2 h-score), despite FGSM producing significantly more perceptible perturbations. On SUN, FGSM has a slight lead over DeepFool of around 1.5 points, though the resulting perturbations of DeepFool are significantly less perceptible. On CUB and SUN, DeepFool takes approximately the same time to produce the attack regardless of the maximum iteration value, suggesting that it manages to cross the boundary in fewer iterations. This is sensible as CUB and SUN has more classes than AWA2, which means class boundaries are closer to each other and thus potentially easier to cross. However, we do not see that effect for C&W, which means it still needs more iterations to cross the boundary despite requiring the largest compute time. On AWA2, FGSM has a decisive lead and DeepFool is somewhat effective. C&W, on the other hand, fails to introduce any meaningful effect, especially in unseen accuracy scores. As one can see in Table 3.1, this is a wider phenomenon; there is a clear discrepancy in how unseen and seen classes are affected. We investigate this phenomenon in the upcoming sections.

Defenses. We first apply the defenses *without* applying any attacks to see the effects of defenses; a defense that degrades the results is naturally not suitable for use. The results are shown in Table 3.2. In this table, SpS, LbS and TVM are spatial smoothing, label smoothing and total-variance minimization, respectively.

We see modest detrimental effects of defenses across the board, which we believe to be acceptable given the improvements they bring. We observe that in AWA2, label smoothing improves the GZSL performance compared to the original ($\approx$ 1 point increase in h-score). There is no clear winner here, although label smoothing and total-variance minimization do a better job than spatial smoothing. We then analyze the effects of each defense under various attack settings. We note that we omit one setting for each attack from our defense analysis; they either introduce extreme perturbations ($FGSM_3$) or negligible effects compared to their weaker counterpart ($DeepFool_2$ and $C\&W_2$).

The spatial smoothing results are shown in Table 3.3 [8]. In the *ZSL* setting, we see good recoveries across all datasets. The recovered accuracy values are, as expected, better for weaker attacks. We see similar recovered accuracy values for each DeepFool and C&W settings ($DeepFool_1$ vs $DeepFool_3$, $C\&W_1$ vs $C\&W_3$), in contrast with what we see for FGSM. This is potentially due to the nature of the attacks; the strength of FGSM scales with the coefficient ϵ, whereas maximum iteration for C&W and DeepFool acts like a switch indicating whether the attacks will perform or not. In the *GZSL* setting, the results follow the trends of ZSL. We see negligible recoveries, up to 0.2 points, for C&W and DeepFool on AWA2 in unseen accuracy values, although its performance for harmonic score and seen accuracy values are better. Surprisingly, spatial smoothing *degrades* the unseen score by 1 point and harmonic scores of $C\&W_1$ by 0.1 points compared to its original (unattacked) accuracy. This will be investigated thoroughly in the following sections.

The *label smoothing* results are shown in Table 3.4. In this table, results denoted with *original* are obtained by training ALE with label-smoothing. Since the model is retrained with label smoothing, we can not compare the recovery performance to Table 3.1. We first note the performance difference between models trained with and without label smoothing (see Table 3.2), and then compare the differences of Tables 3.1 and 3.4. The results show that label smoothing does not introduce a visible improvement against FGSM in both ZSL and GZSL. The results of DeepFool show negligible improvements in harmonic scores on CUB (up to 1 point) and SUN (up to 0.5 points), despite the label smoothing model having lower

[8] Tables 3.3, 3.4 and 3.5 should be compared to Table 3.1.

harmonic accuracy values compared to the original model. The C&W attack goes through the highest recovery rate, especially in the ZSL setting (i.e. up to 10 point improvement on CUB for $C\&W_1$).

The *total-variance minimization* results are shown in Table 3.5. In the *ZSL* setting, all around recoveries for all attack settings are observed. Recovered values for DeepFool and C&W are similar, similar to what we observed for *spatial smoothing*. Among all, TVM does the best job in ZSL (i.e. up to 22 points on CUB for $C\&W_1$). In the *GZSL* setting, similar trends with ZSL are observed. However, we see on AWA2 that unseen accuracies degrade (up to 2.5 points) in the presence of TVM, especially for DeepFool and C&W . For C&W, this decrease is also salient for harmonic scores (up to 2.3 points). As seen in spatial smoothing, TVM has a detrimental effect as well. This will be investigated in the later sections of the thesis.

Summary. In *attacks*, an unbounded, high epsilon FGSM attack is the strongest and the fastest, as one would expect. However, when minimum perturbation is of concern, FGSM loses out to DeepFool and C&W dramatically. Across all datasets, DeepFool exhibits the best trade-off between perturbation magnitude and success rate. In *defenses*, we see varying rates of success for each dataset. On CUB, spatial smoothing is the best for FGSM attacks, whereas TVM is the best for the rest. On AWA2, spatial smoothing is the best across-the-board defense for every attack. For SUN, spatial smoothing is *still* the best for FGSM, however TVM has a lead against C&W and DeepFool. Label smoothing is the worst defense in general and TVM is the most compute-hungry, as expected. We present several qualitative samples in Figure 3.4.

3.8. Analysis

A widely popular notion tells us that adversarial examples are considered as out-of-distribution samples which models fail to recognize. As they do not have their own class prototypes, the ranking system incorrectly assigns them to a class. We require a mechanism to transfer knowledge from clean to adversarial images, on top of the

	Zero Shot			**Generalized Zero Shot**								
	C	S	A	C			S			A		
Attack	Top-1			u	s	h	u	s	h	u	s	h
Original	52.2	55.2	60.6	22.7	56.2	32.4	18.4	31.6	23.3	16.3	74.2	26.8
$FGSM_1$	39.8	46.8	41.1	17.4	43.7	24.9	15.7	25.2	19.4	12.1	59.8	20.1
$FGSM_2$	11.7	15.2	14.1	6.7	9.8	0.80	5.5	7.7	6.4	2.7	10.3	4.4
$DEFO_1$	29.8	30.4	49.6	10.0	20.3	13.4	6.2	8.8	7.3	14.01	42.4	21.1
$DEFO_3$	19.8	19.2	41.7	8.2	11.9	9.7	5.2	7.1	6.0	13.1	25.5	17.3
$CaWa_1$	38.8	45.6	46.6	19.4	40.4	26.3	16.2	26.6	20.1	16.4	61.0	25.9
$CaWa_3$	34.1	42.7	40.6	18.8	34.8	24.5	16.2	25.1	19.7	16.2	52.0	24.7

Table 3.4 Results: *All* images are attacked and defended with *label smoothing*.

	Zero Shot			**Generalized Zero Shot**								
	C	S	A	C			S			A		
Attack	Top-1			u	s	h	u	s	h	u	s	h
Original	54.5	57.4	62.0	25.6	64.6	36.7	20.5	32.3	25.1	15.3	78.8	25.7
$FGSM_1$	49.1	53.2	53.8	23.0	57.5	32.8	18.9	28.3	22.7	11.7	71.8	20.1
$FGSM_2$	25.3	32.8	21.1	12.6	21.9	16.0	12.6	15.3	13.8	5.0	22.5	8.2
$DEFO_1$	48.4	50.3	59.0	19.7	52.3	28.6	15.2	20.8	17.5	12.5	70.9	21.4
$DEFO_3$	48.3	50.3	59.0	19.5	52.3	28.4	15.1	20.8	17.5	12.5	70.6	21.3
$CaWa_1$	50.9	53.3	58.8	24.0	60.3	34.3	20.0	29.2	23.8	12.7	75.6	21.7
$CaWa_3$	51.2	53.4	58.9	24.2	60.2	34.4	19.9	29.1	23.6	12.6	75.6	21.6

Table 3.5 Results: *All* images are attacked and defended with *TV minimization*.

seen-to-unseen transfer we need to tackle already. Furthermore, possibly from a much simpler perspective, ZSL models, especially the ALE model, can be considered as *immature* compared to supervised counterparts; accuracy levels are simply not that high. The second perspective harbors interesting facts. Assuming a model with the perfect accuracy, we know effective attacks can only degrade the results. Effective defenses can degrade the results without any attacks, but we know they alleviate the issues to a certain degree. What happens when the model is far from perfect is exactly what we need to focus on now. Note that this is not a ZSL-specific issue.

Generalized Zero Shot

	C						S						A					
Class Type	U			S			U			S			U			S		
Transitions	CF	FC	FF	CF	FC	FF	CF	FC	FF	CF	FC	FF	CF	FC	FF	CF	FC	FF
$FGSM_1$	70	14	74	49	33	53	51	10	66	49	14	61	83	9	74	33	49	42
$FGSM_2$	96	13	87	98	30	70	98	10	90	98	14	86	100	4	96	92	29	71
$DEFO_1$	87	8	85	81	18	75	93	6	90	95	8	88	46	5	68	51	22	46
$DEFO_3$	93	8	87	92	19	76	94	6	91	98	8	90	52	6	75	70	24	52
$C\&W_1$	86	18	74	75	35	54	52	10	62	52	14	59	76	12	63	37	51	33
$C\&W_3$	92	19	76	85	36	57	56	11	68	59	15	62	87	13	70	54	56	35

Table 3.6 Categorization of prediction changes induced by each attack.

Class-transitions: False/Correct. The attack results show that in the *GZSL* setting, unseen accuracy values are less severely affected compared to seen accuracy values, especially under *weak* attacks. We investigate this by looking at the class-transitions for each attack setting. For each class, we calculate the ratio of class transitions; out of all (originally) correctly predicted samples, what percentage have transitioned to false? Out of all (originally) falsely predicted samples, what percentage have transitioned to correct or *other* false classes? Our results are shown in Table 3.6. In Table 3.6, U and S columns are results for unseen and seen classes. CF, FC and FF are *correct-to-false* (as the percentage of all originally correct predictions), *false-to-correct* and *false-to-other-false* (as the percentage of all originally incorrect predictions) changes in %, represented as ratio averages.

Stronger attacks lead to higher *correct-to-false* (CF) percentages. Moreover, stronger attacks also lead to higher *false-to-other-false* (FF) ratios. This means that regardless of the original predictions, strong attacks induce more class transitions. There is also the possibility of an attack *correcting* an originally incorrect prediction. We observe the highest *false-to-correct* (FC) ratio in C&W (up to 56%) and the lowest in DeepFool (5%). C&W forces false predictions to cross to correct class boundary and fails to push correct classes to incorrect boundaries, which explains its poor performance.

When seen and unseen classes are compared, we observe higher FC ratios for seen classes (i.e. 43 points difference on AWA2 for $C\&W_3$). Furthermore, seen classes tend to have lower CF ratios (i.e. 50 points difference for $FGSM_1$). This is in contrast with the fact that unseen classes are affected less severely in Table 3.1; also see Figure 3.1. However, the original unseen/seen accuracy values of the model are different, especially for AWA2 and CUB. Naturally, the number of originally correct and false predicted samples are wildly different for seen and unseen classes (i.e. unseen accuracy is lower, therefore 10% FC for unseen samples is significantly high in terms of number of samples) and this leads to unseen class accuracy values being affected less severely in absolute numbers, despite being affected more severely in terms of ratios, hence the **pseudo-robustness effect**. This is a byproduct of not having a *strong* model (i.e. low original accuracy). Another interesting fact is the high FC rates in the seen classes; this tells us that in an event of misclassification, the model predicts the correct class with a high probability, but not high enough to be the highest prediction. Therefore, in an attack, since the sample is close to the correct decision boundary, it is highly likely that the sample will be pushed to a close decision boundary, which happens to be the correct decision boundary. This is not the case for unseen classes, which suggests that seen classes are simply *learned* better, which is natural due to the very definition of the ZSL setting. In the same spirit, one can expect high CF ratios for seen classes, however this is not the case. This underlines that the model is robust for seen classes in the case of correct predictions, but its false predictions are not as confident, which is another expected behaviour due to the nature of ZSL.

Class-transitions: Seen/Unseen. We now focus on the effect of attacks from a seen/unseen class perspective. For each class, we calculate the following for all samples and average it for seen and unseen classes: out of *all* changed samples, what percent went to a seen or an unseen class? Our results are shown in Table 3.7. In Table 3.7, UU, US, SU and SS are unseen-to-unseen, unseen-to-seen, seen-to-unseen and seen-to-seen transitions, respectively.

The results show that except FGSM, attack characteristics in terms of seen/unseen class transitions seem stable. For FGSM, we see increase in unseen-to-seen transitions, which is in line with the further decrease of accuracy values (i.e. unseen-to-unseen can have false to correct transitions for unseen). This behaviour is in agreement with the attack settings;

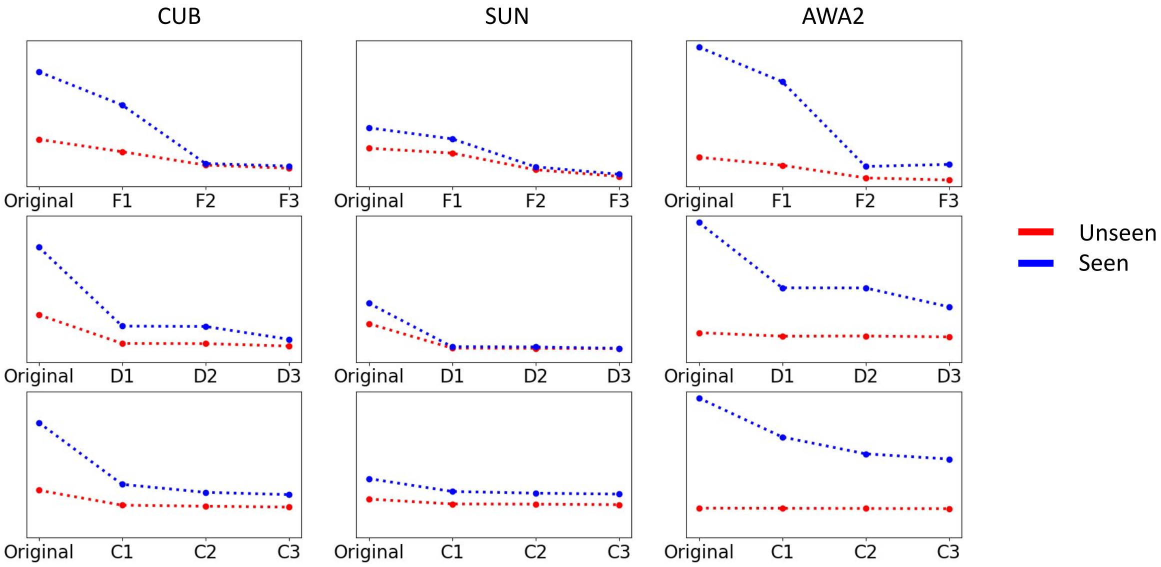


Figure 3.1 The effects of FGSM (F), DeepFool (D) and C&W (C) attacks.

FGSM strengthens its attack with higher ϵ whereas DeepFool and C&W simply have more time to *solve* for the minimum perturbation with higher iterations. We also see that originally seen classes are more likely to go to a seen class, compared to an originally unseen class (i.e. SS values are higher than US values), which gives us hints about the class decision boundaries. Regardless of the dataset and the attack, an overwhelming majority of the transitions happen towards seen classes. We hypothesize the reason for that to be two-fold. First, the number of seen classes are higher than that of unseen classes, with varying degrees, for each dataset. This only does not explain the decisive tendency towards seen classes, however. Second, and more importantly, the model trains exclusively on seen classes and, as expected, is more confident about its predictions, and this causes a severe bias towards seen classes in the boundary transitions. See Figure 3.2 for a visualization of the seen/unseen class transition trends.

Adverse effects of defenses. As shown in Section 3.7., there have been cases where defenses reduced the accuracy after the attacks rather than recovering it (also see Figure 3.3). Following the work shown in Table 3.6, we observe the effect of defenses (i.e. we add another layer to CF, FC, FF transitions, such as CFC, FCF, FFC, etc). We can analyse the effect of

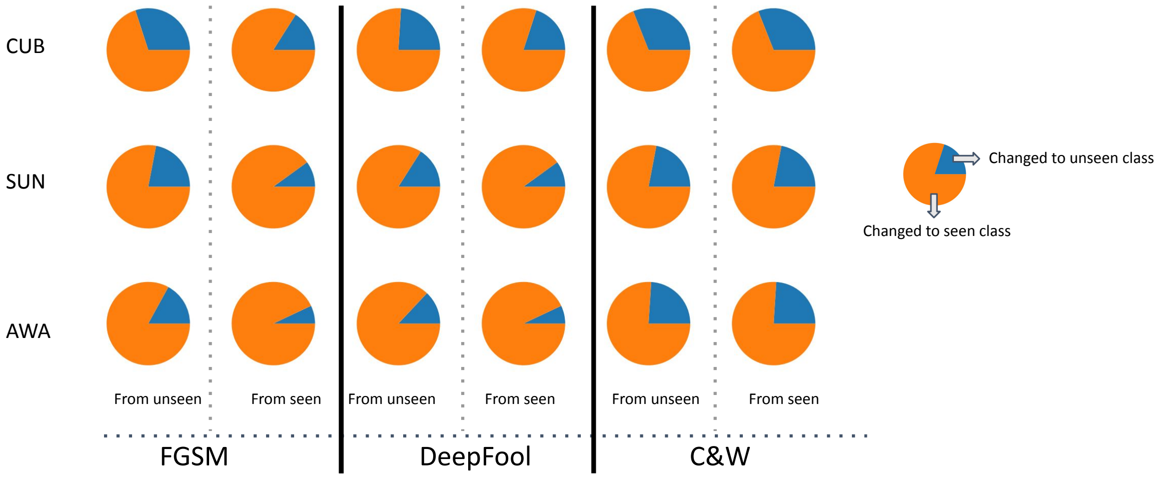


Figure 3.2 Seen and unseen class transition charts for all attacks and datasets.

	Generalized Zero Shot											
	C				S				A			
Class Transition	UU	US	SU	SS	UU	US	SU	SS	UU	US	SU	SS
$FGSM_1$	30	70	16	84	22	78	10	90	17	83	7	93
$FGSM_2$	28	72	18	82	17	83	10	90	12	88	7	93
$DEFO_1$	24	76	20	80	16	84	10	90	13	87	7	93
$DEFO_3$	24	76	20	80	16	84	10	90	14	86	7	93
$CaWa_1$	31	69	17	83	22	78	10	90	24	76	8	92
$CaWa_3$	31	69	17	83	22	78	10	90	25	75	9	91

Table 3.7 Per-class normalized class transitions (in %) for different attacks.

defenses in four major categories; correcting a mistake (CFC, FFC), preserving the results (CCC, FFF) and having detrimental effects (CCF, FCF) and failure to recover (CFF, FCC). It must be noted that recovery here means recovering the *original* label, not the correct label. Across all experiments, we observe every category of effect, with correct-recoveries (CFC) leading the overall recovery of accuracy. However, we observe that the defense-induced reduction of accuracy values correlate well with high FCF ratio. This effectively means that the defense is simply negating the *positive* effect of attacks; the defense does its job well by recovering the original predictions, but the original predictions may simply not be correct.

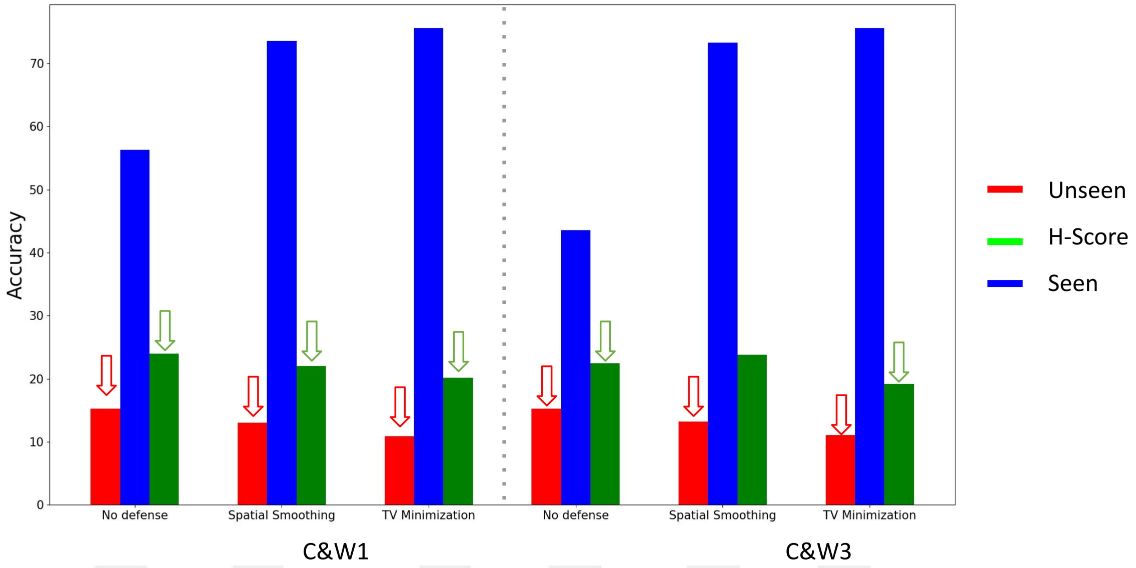


Figure 3.3 TVM and SS defenses *adversely* affecting accuracy.

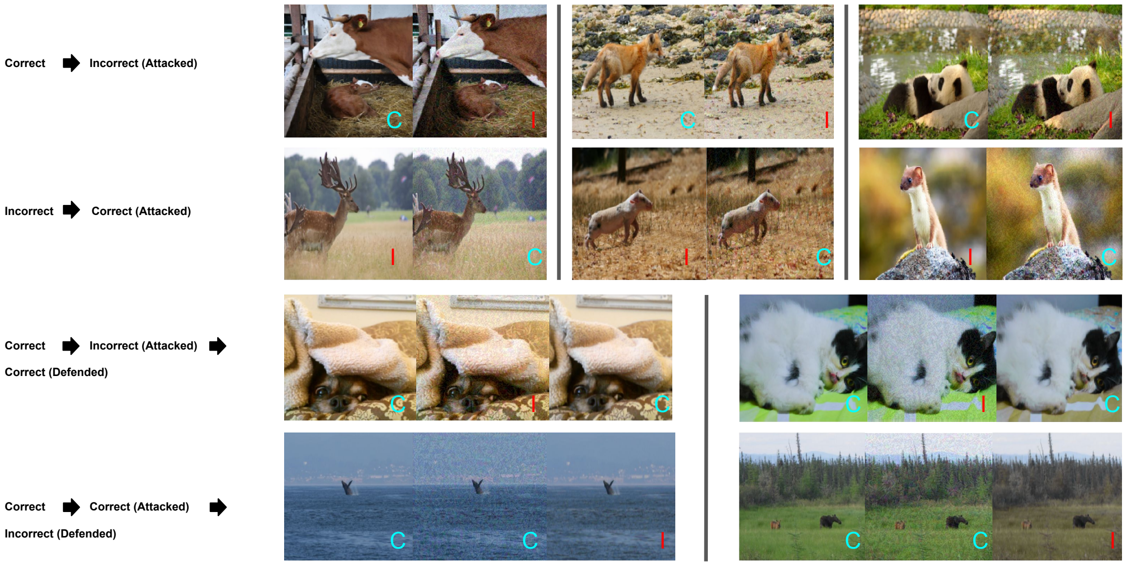


Figure 3.4 Various class transitions induced by attacks and defenses.

Attacking only correct predictions. We have seen interesting trends so far, such as defenses degrading the accuracy values, attacks increasing accuracy values and unseen accuracy values being less severely affected despite the results of Table 3.6. Our model is far from being perfect, therefore it is imperative to decouple potential effects of low accuracy from ZSL-specific trends. We then attack only the correct predictions towards this end. We see that the *unintuitive* effects such as attacks *improving* the results or defenses *degrading* the results are gone. Across all attacks, we see significant accuracy reductions and we see improvements across all defenses. The overall *rankings* for best attack and defense follow our previous *all images* attack settings, therefore we do not include detailed numerical results. In this setting, results are slightly more reminiscent of a supervised model. However, the trends shown in Tables 3.6 and 3.7 are still very much valid; unseen classes are affected more severely and class transitions happen overwhelmingly towards seen classes. This suggests the extreme class imbalance inherent in ZSL has a significant impact on robustness behaviour.

ZSL vs GZSL. The average performance of the model under ZSL and GZSL evaluation show varying results; on CUB, harmonic scores are affected more severely than ZSL accuracy. On SUN, both are effected quite similarly whereas on AWA2, ZSL accuracy is impacted more heavily. The defenses work visibly better for GZSL than ZSL, with the only exception being AWA2 with label smoothing defense.

Dataset characteristics. The datasets under consideration are wildly different in their characteristics; SUN and CUB have fewer samples per class and high number of classes, whereas AWA2 has high number of samples per class but fewer classes. On AWA2, we see attacks failing to effect in their weakest setting; DeepFool and C&W introduce 2 and 0.1 point reduction, respectively. We observe that FC transitions happen more frequently on AWA2 (up to 56%) compared to others; this is potentially linked to having multiple confident predictions as this is more prominent for seen classes. Upward accuracy spikes happen more frequently on AWA2 as well (especially in the ZSL setting); this is likely an effect of having fewer number of classes, as misclassifications are simply more likely to fall into the originally correct class. Transitions to unseen classes occur rarely on AWA2 (with a maximum of 9% for $C\&W_3$). Finally, we see the discrepancy between seen and unseen classes to be pronounced on AWA2. All these insights support the claim of a larger distribution helping robustness [189], since AWA2 is larger and has more per-class samples compared to others. We see similar trends for SUN and CUB; SUN has the fewest transitions to unseen classes.

This correlates strongly with the high number of classes in SUN. In summary, we see SUN and CUB getting better returns from all defenses, compared to AWA2.

3.9. Conclusion and Future Work

Despite their impressive rise, it is shown that machine learning models can be fooled with carefully designed perturbations. Adversarial robustness have been primarily studied from a supervised model perspective. ZSL and GZSL algorithms that lack supervision for a set of classes have not received attention for their adversarial robustness. In this chapter, we introduce a study that aims to fill this gap by evaluating a well-known ZSL model for its adversarial robustness, both in ZSL and GZSL evaluation set-ups. We expose the model to numerous attacks and defenses across popular ZSL datasets. Our results show that adversarial robustness for ZSL has its own challenges, such as the extreme data bias and the immature state of the field (compared to supervised learning). We highlight and analyse several key points, especially in GZSL settings, to guide future researchers in what needs attention in making ZSL models robust and also what points could be of importance for interpreting the results. Finally, we identify and discuss the *pseudo-robustness effect* often observed in our work, where absolute metrics may not always reflect the robustness behaviour of the model.

4. Unifying Adversarial and Corruption Robustness for Discriminative ZSL

In this chapter, expanding on the previous chapter, we focus on common image corruption robustness of discriminative ZSL models. We highlight some key points of this chapter; i) three new datasets for robustness analyses in ZSL, ii) new strong baselines, iii) expansion of selected ZSL models, iv) thorough analyses of corruption robustness behaviour and v) a detailed comparison with adversarial robustness results of chapter 3..

> *'The corruption of the best things gives rise to the worst.'*
>
> *David Hume*

4.1. Introduction

Adversarial machine learning, with various attack [100–102, 104, 165, 167] and defense [30, 116, 118, 120, 122, 123, 125, 174] methods, brought a fresh perspective to robust generalization [190]. Despite the discussions stemming from adversarial ML, adversaries are *worst-case* scenarios and tend to require *bad intent* to materialize. There are other effects which do affect images and the accuracy of a model. A subset of these effects, called *corruptions*, occurs more frequently and naturally, are not *worst-case* scenarios and are not necessarily imperceptible. [9] defines a variety of corruption categories, and shows that they can invalidate otherwise state-of-the-art models in supervised regimes.

A large part of the literature in robustness has focused on supervised models. Zero-Shot Learning (ZSL) and Generalized Zero-Shot Learning (GZSL) [64, 185] differ from fully supervised settings; in ZSL, the aim is to learn from a set of classes such that the model performs well on classes unseen during the training. GZSL extends this such that the model performs well on both seen and unseen classes. As mentioned in the introduction of

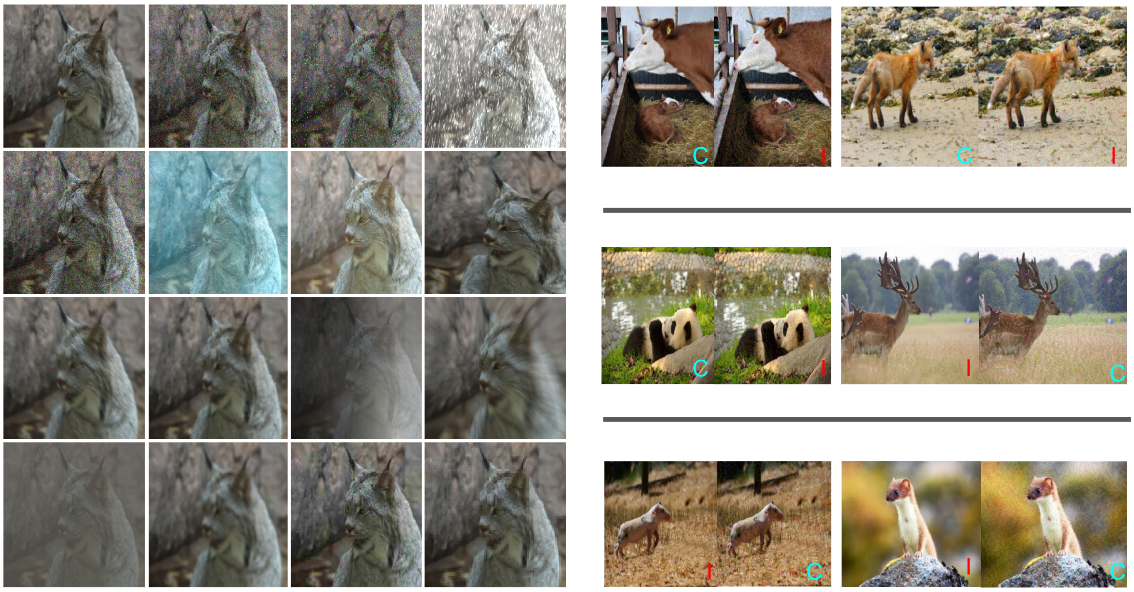


Figure 4.1 Examples from our datasets (left) and class transition examples.

chapter 3., ZSL serves as an interesting, and particularly challenging, medium for robustness analyses. Therefore, we continue our focus on discriminative ZSL models.

In this chapter, we extend our previous work on adversarial attacks (chapter 3.) and present in-depth analyses on the robustness of discriminative ZSL models from a common image corruption aspect. We leverage the well-established, discriminatively trained label embedding model [83, 166]. We then extend our analyses to a different family of ZSL models with the attribute attention model (LFGAA) [1], and subject both to corruptions and corruption-defenses. We also curate and publicly release the first corruption benchmarks for ZSL/GZSL, with the names CUB-C, SUN-C and AWA2-C, with example images shown in Figure 4.1. Jointly based on our new observations regarding natural corruptions and those from chapter 3. regarding adversaries, we conclude our study with important insights and discussions on dataset characteristics, class boundary transitions, severe class imbalance and discrepancies between ZSL and GZSL performances. In summary, our contributions are listed as follows.

- We present a large set of experiments (over 1000+) focusing on the robustness of discriminative ZSL models from a corruption aspect. To the best of our knowledge, this is the first study to establish such a benchmark.
- We curate and release the first benchmark datasets for corruption analyses in ZSL; CUB-C, AWA2-C and SUN-C. The code and datasets are available at `https://github.com/MKYucel/zero_shot_corruption_benchmarks`.
- Our results show that discriminative ZSL models are not robust at all, and we hypothesize the reasons to be severe class imbalance and model weakness. Combined with the results of chapter 3., we show that the *pseudo-robustness effect*, where absolute metrics may not always reflect the robustness behaviour of a model, is present for adversarial attacks and not for corruptions. This *pseudo-robustness effect* is visualized with examples in Figure 4.1 (right-side images).
- We show that several defense methods improve the clean accuracy and set new strong baselines for both label-attribute embedding and attribute attention models.
- We show in detail that unseen and seen classes are affected disproportionately by corruptions. We also show zero-shot and generalized zero-shot performances are affected differently as well.

This chapter is structured as follows. In Section 4.2., we review the literature on ZSL, corruption robustness and robust generalization. In Section 4.3., we motivate our work by presenting the model selection process, dataset creation and methods we use to create our benchmark. In Section 4.4., we show our experimental results and analyses. We perform a concluding comparison of model robustness under adversaries and corruptions in Section 4.5.. We conclude with our final remarks in Section 4.6..

4.2. Related Work

Robust generalization. Adversaries tend to require malicious intent and have arguably low probability of occurrence. Therefore, for a stronger analysis of robust generalization, additional venues are essential. It has been shown that ImageNet-trained CNNs are biased towards texture, and this can be partially alleviated with a new benchmark Stylized-ImageNet, formed of images with conflicting textures and shapes [8]. Another work released ImageNet-A, which is a curated collection of notoriously hard examples of ImageNet classes [95]. The effects of distribution shifts have been analysed with various datasets, such as ImageNet-R, SVSF and DeepFashion Remixed [33]. For a detailed survey, see chapter 2..

Corruption robustness. Corruption and perturbation robustness for various ML models have been discussed in [9, 132], where ImageNet-C, ImageNet-P, CIFAR-C and MNIST-C benchmarks have been proposed. The corruption robustness datasets simulate common image corruptions, such as noise, weather, blur and digital degradations on various severity values. In these works, it has been shown that virtually all state-of-the-art models are invalidated when exposed to these corruptions. Several studies showed that data augmentation techniques can help with robustness [33, 36, 142]. It is also shown that adversarial training [144], self-supervised learning [143], arbitrary style-transfer [148], adversarial noise-training [145] and rectified batch normalization [149] helps improve corruption robustness. Corruption robustness is an active field with a large potential for improvement.

Zero-shot Learning. ZSL aims to facilitate learning under severe class imbalance, where the model is trained on a subset of classes and is required to perform accurately across all classes, even for the ones unseen in training. ZSL models do this by exploiting intermediate auxiliary information, commonly in the form of attributes, to transfer knowledge between seen and unseen classes. Since we extensively covered this topic in chapters 2. and 3., we refer the readers to the respective chapters for more information. For detailed surveys on the topic, see [55, 185].

A preprint [186] proposes an adversarial training regime to train ZSL models robust to adversaries. Our study, in contrast, focuses on the intrinsics of discriminative ZSL models from a corruption robustness aspect. Combined with the previous work presented in chapter 3., we present a thorough analysis of discriminative ZS robustness with different experimental setups where we assess not only the effect of corruptions, but also draw insights on dataset characteristics, class transitions and ZSL/GZSL discrepancies. We also present and release the first benchmark datasets for ZSL corruption robustness.

4.3. Methodology

In this section, we provide details on our ZSL-robustness benchmarks, present the defense methods and ZSL formulation used in our analyses.

4.3.1. Benchmarking ZSL corruption robustness

The goal of *corruption robustness* is to analyse and understand scenarios where naturally occurring image degradations harm the performance of a model. Therefore, corruption robustness is of great practical importance for real-world use cases. It is imperative to have a standardized representation of these effects so that common test beds can be constructed to facilitate principled progress in the field.

A key contribution of this work is the curation and release of the corrupted versions of three ZSL datasets; *CUB-C, SUN-C and AWA2-C*. In generating these benchmarks, we largely follow the principles of the ImageNet Corruption (ImageNet-C) [9] dataset due to several reasons. First, we believe that the corruption types of ImageNet-C sufficiently cover the possible corruption effects an image may experience. Second, many ZSL methods use features from ImageNet-pretrained models (*i.e.* ALE) or finetunes ImageNet-pretrained backbones (*i.e.* attribute-attention model). We hypothesize it is only logical to maintain parity with ImageNet-C corruption types to evaluate the robustness of the ZSL methods, as well as the underlying representations they use.

The proposed *SUN-C, CUB-C and AWA2-C* datasets have four corruption categories (weather, digital, blur and noise) and consist of 15 corruption types with 5 severity levels each. We use the same corruption types in ImageNet-C [9]; gaussian, shot and impulse noise for *noise* category; glass, defocus, motion and zoom blur for *blur* category; snow, frost, fog and brightness for *weather* category; contrast, elastic transform, pixelate and jpeg compression for the *digital* category.

We adapt the existing corruptions to be suitable to the characteristics of our ZSL datasets (i.e. image sizes, semantics, etc.) and ensure they are representative of common detrimental effects (*i.e.* we make sure they are not too weak or too powerful). Finally, due to the vast storage requirements, we corrupt the images on-the-fly in a deterministic way to achieve reproducibility. See Figure 4.1 for example visuals.

4.3.2. Corruption robustness baselines

We also assess numerous defense methods for their ability in combating the adverse effects of corruptions. We first assess three adversarial defense methods; label smoothing [187], total variance minimization (TVM) [123] and spatial smoothing [124] [9]. Additionally, we experiment with two methods which are effective against image corruptions: AugMix [36] and ANT [145].

AugMix leverages two simple building blocks; existing augmentation operations and a consistency loss. The augmentation operations do not overlap with ImageNet-C corruptions for fair analyses. AugMix samples up to three augmentation operations with different severity values and creates an *augmentation chain*; these augmentations are to be applied successively. The algorithm can create up to three augmentation chains that will work in parallel. The final mixing operation, where the results of the chains are combined, is performed with randomly sampled weights for each branch. The resulting image is combined with the input image with randomly sampled weights, and the very final image is obtained. This process is often done twice in parallel, giving us three images; the original image, augmented image one and augmented image two. Finally, these three variants are used to enforce a JS-divergence consistency loss (i.e. encoding these three images into similar

[9] Readers are referred to chapter 3. for further details of these techniques.

embeddings) based on the fact that their semantics (i.e. class labels) remain more or less the same.

In contrast with augmentation methods that use existing corruptions or their combinations, ANT makes one of the first attempts to learn an additive noise augmentation to optimize the robustness of the model. ANT first trains a generative network against a trained classifier (i.e. ImageNet classifier) to generate the adversarial noise. Note that this part is essentially similar to existing adversarial example methods that learn how to generate adversaries. ANT takes one step further and then jointly trains the classifier and the noise generator; it essentially finetunes the classifier on images with generated noise added to them.

4.3.3. Zero-shot learning model

Following the practice of chapter 3., we use the well-known label-embedding formulation (ALE) [83], which has been shown to be a stable and competitive method even in modern benchmarks [185]. Although we have produced a detailed analyses in chapter 3., we find the results to be somewhat limiting as we focus on a specific family of discriminative ZSL models; ALE is an inductive model that does visual to semantic mapping with a linear compatibility function. It is obvious that experimenting with every ZSL models is not tractable, therefore we aim to hit a sweet spot between rigorousness and tractability.

To this end, we select the attribute attention model LFGAA [1]. LFGAA brings about several advantages to our analyses; i) it is a more recent model that is significantly more accurate than ALE, therefore its inclusion will let us assess how much of a factor model strength plays in ZSL robustness, ii) it hails from a different family of models, that does a non-linear mapping, via backbone finetuning, to both semantic and latent feature spaces, and iii) it has an inductive and a transductive variant (LFGAA+SA), which will increase the depth of our analyses.

LFGAA identifies the problem where all attributes are treated equally with respect to their discriminative power; i.e. spotty pig and dalmatian dogs are given as examples where both have similar attributes in their *spottiness*, which might cause misclassifications. LFGAA addresses this by an object-based attribute attention, where attributes are attended with respect to the objects in the image. LFGAA, similar to ALE, learns the compatibility score defined as

$$F^{'}(x, y; W) = \theta(x)^T W diag(p(x))\phi y \tag{21}$$

where $p(x)$ is the proposed object-based attention, W is the learnable parameters used for projection (i.e. the network), $\theta(x)$ is the visual embedding and $\phi(y)$ is the class-embedding (i.e. attributes). The transductive variant of LFGAA (LFGAA+SA) uses a *self-adaptation* mechanism to pseudo-label unlabeled samples and then iteratively refines the prototypes. This process lets the method leverage additional unlabeled data.

We acknowledge that especially in the GZSL setting, most state-of-the-art methods are generative approaches, where unseen class samples are generated using class embedding conditional generative models [76, 78, 79, 184]. These methods require complex pipelines where synthetic samples are generated and models are expected to learn from synthetic samples and perform well on real samples. These methods present interesting venues for robustness analyses, but their complex nature complicates the robustness analyses. Therefore, we note that we keep the generative approaches outside our scope and focus primarily on discriminatively trained ZSL models.

4.4. Experiments

In this section, we present the experimental setup and implementation details, and then show our results and analyses.

4.4.1. Datasets and evaluation metrics

Similar to chapter 3., we train our models on CUB, SUN and AWA2 datasets. CUB is a mid-sized dataset with 312 attributes, 200 classes and 11788 images. As classes are similar to each other appearance-wise and each class has few samples, it represents a challenging case. SUN is another mid-sized dataset that has 102 attributes, 717 classes and 14340 images. Due to having significantly different classes and even lower per-class sample compared to CUB, SUN is also a quite challenging dataset. AWA2 is a larger dataset with 85 attributes, 50 classes and 37322 images. It has a good per-class sample count, which leads to a more severe class imbalance between seen and unseen classes, and often lead to overfitting in GZSL settings. We use the splits proposed in [185]. For corruption evaluations, we use our newly proposed datasets *SUN-C, CUB-C and AWA2-C*, and evaluate our models using the previously mentioned 15 corruptions with 5 severity levels.

In the corruption robustness literature, *Mean Corruption Error* (MCE) is the common metric; MCE calculates the errors for each corruption type using all five severity levels and weighs them using AlexNet errors, and then calculates the mean of each corruption category to produce the MCE score [9]. Due to the sheer scale of experiments, providing MCE values is not feasible [10] for us. Instead, we calculate the corrupted accuracy values for each corruption type and severity level, find the reduction in accuracy compared to the original accuracy, and then calculate the ratio of reduction (i.e. as a percentage of the original non-corrupted results) for each corruption category [11], as shown in Figures 4.2 and 4.5. For defenses, we do the same, but with a key change; instead of finding the reduction in accuracy compared to the original accuracy, we find the reductions in accuracy (produced by the defense method) compared to the corrupted accuracy values (for each severity and type) and then calculate the ratio of reduction (i.e. as a percentage of the original non-corrupted results) for each corruption category, as shown in Figures 4.3, 4.4, 4.6 and 4.7 [12]. We hope that our results will be the *ZSL-version* of AlexNet error-weights for ZSL corruption robustness, due to the seminal nature of ALE model.

[10] We provide MCE values in the appendix A.

[11] Our results are reminiscent of Relative MCE [9], which is another metric for corruption robustness.

[12] We note that Figures 4.3 and 4.4 can be compared to each other but not to Figure 4.2.

4.4.2. Implementation details

We merge ResNet-101 [4] feature extractor with ALE. We train the ALE model, keeping the feature extractor fixed. For LFGAA, we retrain all the models on all datasets for both inductive and transductive variants. We use ResNet backbone for CUB and SUN, and VGG for AWA2. We indicate the reproduced values as *LFGAA+SA* and *LFGAA+Hybrid* for LFGAA, and *original* for ALE, although there are slight variations coompared to original results [1, 185]. We use PyTorch [188] for our experiments.

We set 0.9 to the ground-truth label for label smoothing. For spatial smoothing, we use 3x3 windows. The maximum iteration is 3 for TVM. We apply the same corruption and defense parameters to all datasets for a fair comparison. We first resize and then corrupt the images to make sure a comparable effect is achieved on all images. We leverage the original implementations for AugMix (with JS divergence) and ANT, and tune the results on our validation corruptions.

In the figures across this chapter, *N, D, B* and *W* are *noise*, *digital*, *blur* and *weather* categories, respectively. Blue, red, orange and green bars indicate ZSL top-1, unseen, seen and harmonic scores, respectively.

4.4.3. Corruption robustness experiments - ALE

Corruptions. The corruption performance of ALE is shown in Figure 4.2. In the *ZSL* setting, we see all-around reductions of accuracy values. On AWA2, the highest ZSL reduction is in the blur category. On CUB, the noise category introduces a significant reduction in accuracy values up to 60%. SUN undergoes the highest reduction in accuracy (60%) when exposed to noise. For all datasets, digital corruptions are the weakest. On individual corruption types, we observe brightness to be the weakest link. Gaussian noise, shot noise and contrast corruptions produce the highest accuracy drops on CUB, SUN and AWA2 datasets, respectively.

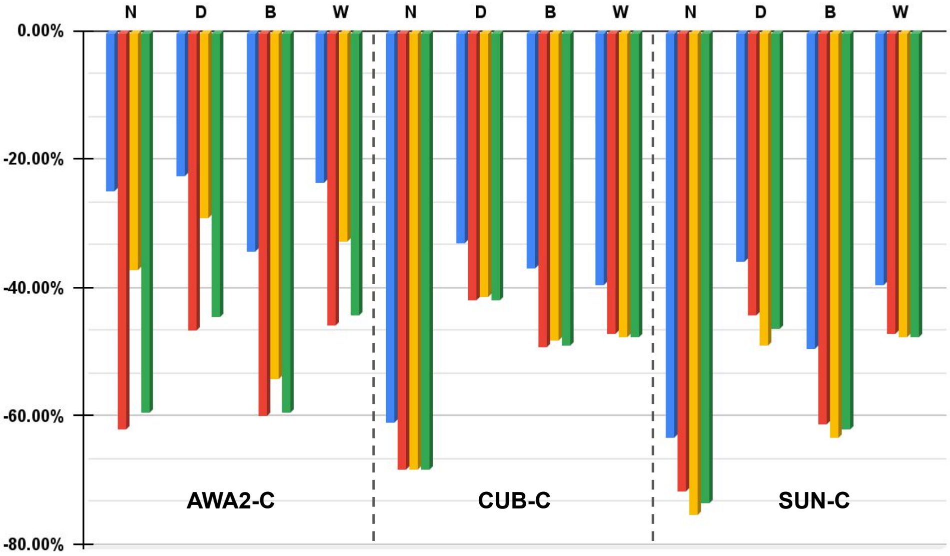


Figure 4.2 Category-based reductions, as a % of the clean results.

In the *GZSL* setting, corruptions introduce decisive reductions across all datasets. On AWA2, noise and blur introduce around the same degradation in harmonic scores (60%), although effects on unseen and seen classes for noise and blur do vary. On CUB and SUN, noise leads to the highest reduction in harmonic scores (up to 75% on SUN). On AWA2, unseen classes suffer more severely compared to seen classes. On CUB, seen and unseen classes are affected somewhat similarly, whereas on SUN seen classes are affected slightly more. On individual corruption types, brightness is still the weakest one. Contrast, impulse and Gaussian noise corruptions produce the largest accuracy drops on AWA2, CUB and SUN, respectively.

Defenses. The spatial smoothing results are shown in Figure 4.3. In the *ZSL* setting, spatial smoothing adversely affects the results across all datasets. On AWA2, noise is affected the worst (-7.5%) whereas digital is the least affected (-3.5%). On CUB and SUN, weather and noise are the worst and least affected, respectively. Spatial smoothing seems to work only on SUN dataset for noise (+1.5%) corruptions. On individual corruption types, spatial smoothing only works well against impulse noise for all datasets. Spatial smoothing does not work for *GZSL* either. Across all datasets, weather and noise are the worst and the least affected, respectively. The only time spatial smoothing works for GZSL is for noise on SUN dataset again; it improves unseen classes and harmonic scores (0.5%). Reminiscent of ZSL, spatial smoothing only works against impulse noise for all datasets in GZSL.

The *total variance minimization* results are shown in Figure 4.3. In the *ZSL* setting, TVM fails to introduce any improvement, but actually is better than spatial smoothing. Across all datasets, we see the noise results either improved (*i.e.* +3.4% on SUN) or affected negligibly whereas weather category is affected the most. We see consistent recoveries for noise on CUB and SUN, and also blur shows improvements on SUN. On individual corruption types, we see all noise types undergo positive trends in all datasets. On AWA2, some blur types are recovered as well, especially in high severity levels. On CUB, all noise types and glass blur are improved. On SUN, all-around improvements are seen for all noise and blur corruptions. In the *GZSL* setting, the results are worse, but still better than spatial smoothing. Across all datasets, weather is the worst affected one (*i.e.* -10% on AWA2 h-score) again. Noise and blur are the best for AWA2/CUB and SUN, respectively. We see positive trends only in noise and blur for CUB and SUN, respectively. We observe unseen class accuracy levels leading the recoveries when there is an actual improvement (*i.e.* +2.5% in unseen on SUN for blur). On individual corruption types, we see zoom blur to be the only corruption that is improved on AWA2. All noise types and glass blur experiences recoveries on CUB. On SUN, all blur types are improved. Unseen classes still experience less degradation than seen classes, specifically on SUN and AWA2 datasets.

For label smoothing, AugMix and ANT, we retrain the models and show the results in Table 4.1. Label smoothing has a slight negative effect, except on AWA2 GZSL where it actually improves the unseen (+1 point) and harmonic scores (+1.1 point). AugMix performs worse than label smoothing in ZSL, but it improves ZSL top-1 on SUN dataset by 1 point. In GZSL, AugMix successfully improves seen, unseen and harmonic scores, except on CUB where seen class accuracy is reduced by 4.5 points. ANT has a similar trajectory as AugMix; it degrades ZSL scores. In GZSL, it improves unseen classes for all datasets and improves harmonic scores on AWA2 by around 6 points and SUN by 3.3 points, **which leads to improved strong baselines**.

	Zero Shot			**Generalized Zero Shot**								
	C	S	A	C			S			A		
	Top-1			u	s	h	u	s	h	u	s	h
Original	**54.5**	57.4	**62.0**	25.6	**64.6**	36.7	20.5	32.3	25.1	15.3	78.8	25.7
LbS	52.2	55.2	60.6	22.7	56.2	32.4	18.4	31.6	23.3	16.3	74.2	26.8
AugMix	51.6	**58.4**	54.9	**27.2**	60.1	**37.5**	**23.6**	35.7	**28.4**	16.1	83.5	27.0
ANT	48.9	57.4	55.1	26.1	60.6	36.4	**23.6**	**35.8**	**28.4**	**19.3**	**85.4**	**31.6**

Table 4.1 Results of ALE models trained with *label smoothing*, *AugMix* and *ANT*.

The label smoothing results are presented in Figure 4.3. In *ZSL*, we see negligible improvements in isolated cases (*i.e.* weather category on CUB, +0.3%). On AWA2, all categories are affected similarly but weather is the worst affected (-5%). On CUB, weather results are improved slightly, and noise is the worst affected one. On SUN, all categories reduce the accuracy slightly. In *GZSL*, we see improvements, especially on AWA2. Unseen and harmonic scores are recovered quite visibly (up to 12%), possibly due to the increase in unseen values shown in Table 4.1, even though seen accuracy values do get worse. Similar to AWA2, weather enjoys the best results on CUB, even though it experiences accuracy drops. On SUN, we see minor improvements in seen accuracy values for digital (0.2%) and weather (1.5%), whereas unseen accuracy values go through significant degradation. On individual corruption types, the trends follow the category-wise results, therefore no further detail is provided.

The AugMix results are shown in Figure 4.4. In *ZSL*, AugMix simply fails to deliver. Noise categories are the worst affected ones, except digital which is a bit worse than noise on CUB. Blur category enjoys the smallest performance drop across all datasets. Despite not seeing improvements in ZSL accuracy levels, the adverse impacts are visibly weaker than the previous defenses; on SUN, blur and digital barely introduce further degradation (-0.2% for both). In the *GZSL* setting, we see decisive improvements across all datasets. Except seen classes on CUB, where reductions up to 5% are observed, both seen and unseen classes are recovered all-around. Except some isolated cases (digital and weather on AWA2), unseen class recoveries are visibly better than seen classes. Compared to label smoothing,

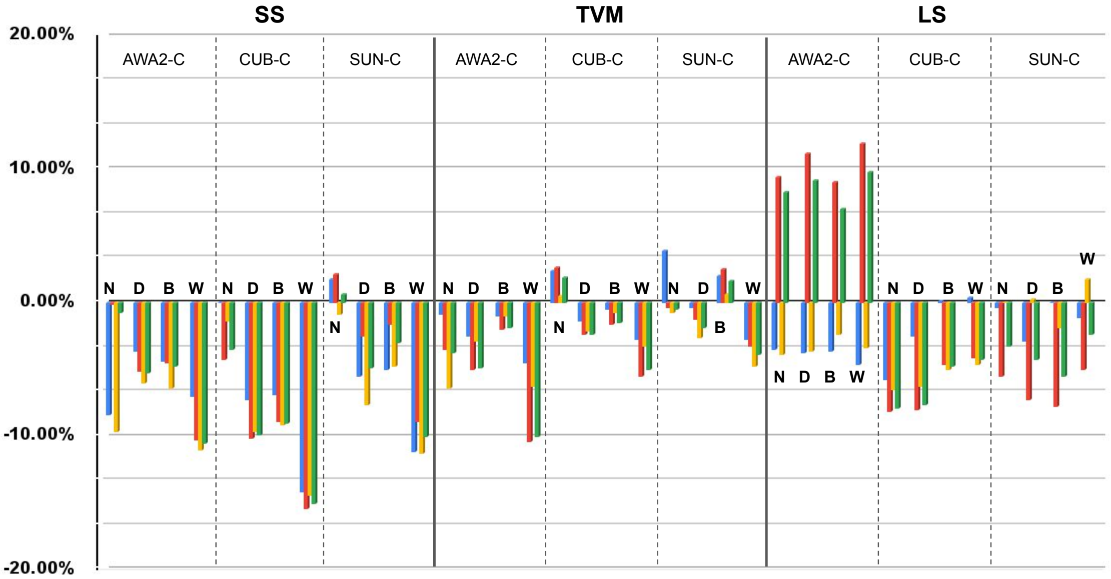


Figure 4.3 SVM, SS and LS defense performance, as a % of clean results.

AWA2 recoveries are slightly worse but AugMix is effectively the first method that provides consistent accuracy recoveries. Noise and weather categories undergo the highest recovery rates for AWA2 (+18% in h-score) and SUN/CUB (up to +5% in h-score), respectively. On individual corruption types, the trends are similar to category-level results, therefore we do not provide further details.

The results of ANT are shown in Figure 4.4. In *ZSL*, ANT performs better than all others. The results show consistent recoveries on SUN, partial recoveries on CUB and only slight degradations on AWA2. On AWA2, noise performs the best whereas weather is the worst (-7%). Blur goes through accuracy drops (-4%) and slight improvement (+1.2%) on CUB and SUN, respectively. Noise performs the best and introduces significant improvements in both CUB (+3%) and SUN (+10%). In the *GZSL* setting, we see clear and consistent accuracy recovery performance, which makes ANT comfortably the best performing defense method among all we have tested. Across all datasets and noise categories, unseen and harmonic accuracies are improved. Seen classes, except digital and blur on CUB (-5%), undergo recoveries as well. Unseen/seen discrepancy is quite visible, and unseen classes

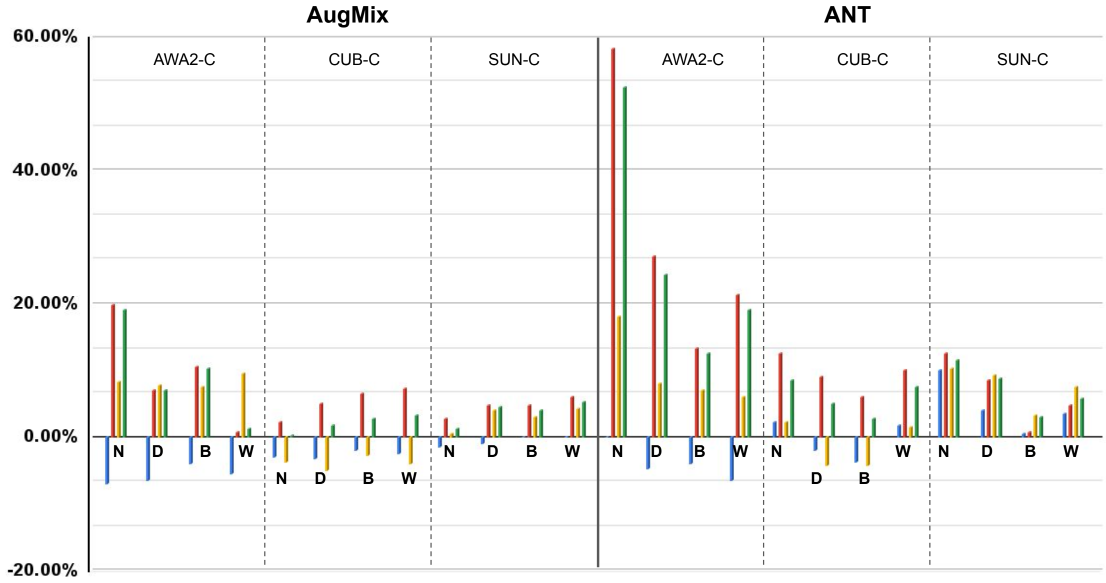


Figure 4.4 AugMix and ANT defense performance, as a % of clean results.

often enjoy better returns with ANT.

Summary. We observe that the digital corruptions are basically the weakest ones. Noise categories introduce the most dramatic accuracy reductions, where accuracy drops of 60% on AWA2 h-score and 73% on SUN h-score are observed. Brightness corruption is the weakest one all-around, whereas different noise types and surprisingly contrast (a digital corruption) are the most effective corruption types. In defenses, ANT is the most successful defense, whereas spatial smoothing seems to be the most ineffective one.

4.4.4. Corruption robustness experiments - LFGAA

Corruptions. The corruption results of LFGAA is shown in Figure 4.5. In the *ZSL* setting, the transductive variant (*LFGAA+SA*) shows all-around accuracy drops, where the reductions are least significant on AWA2. The inductive variant (*LFGAA+Hybrid*) looks more robust compared to *SA*; especially on AWA2 the effect of the corruptions are significantly smaller (-5% in digital). In the *GZSL* setting, both variants experience accuracy drops all-around.

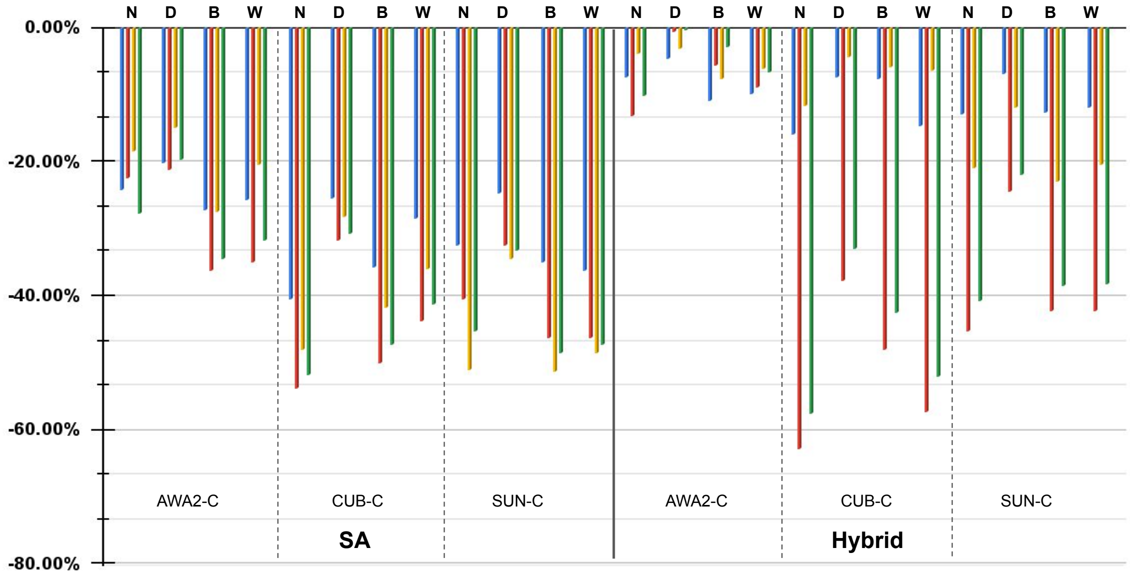


Figure 4.5 Category-based reductions, as a % of the clean results- LFGAA [1].

Except SUN in *LFGAA+SA*, virtually all others show that unseen classes experience a more visible degradation. *LFGAA+Hybrid* shows a slightly less severe degradation than *LFGAA+SA* in the *GZSL* setting, which is quite prominent for AWA2 results. Especially in *LFGAA+Hybrid*, unseen classes are affected severely.

Defenses. We provide the accuracy values of LFGAA trained with AugMix and ANT in Table 4.2, as well as the *reproduced* original results. We see that the effects on AugMix and ANT are a mixed-bag. AugMix improves on CUB ZSL (+2.6%) and AWA2 GZSL (+7.5%) for *LFGAA+SA* and improves on AWA2 GZSL (+5%) for *LFGAA+Hybrid*. ANT performs better compared to AugMix, where it shows improvements on ZSL and GZSL for all datasets except SUN GZSL for *LFGAA+SA*. In *LFGAA+Hybrid*, it shows negligible degradations for ZSL but shows consistent improvements in GZSL, **leading to improved strong baselines**.

AugMix, for *LFGAA+SA* (see Figure 4.6), shows improvements in ZSL except some select categories, especially on SUN. In the *GZSL* setting, except SUN, it shows significant improvements. The results also show that AugMix helps recover unseen class accuracy values better than seen class accuracy values. For *LFGAA+Hybrid* (see Figure 4.7), AugMix

	Zero Shot			**Generalized Zero Shot**								
	C	S	A	C			S			A		
		Top-1		u	s	h	u	s	h	u	s	h
LFGAA+SA	78.9	58.7	74.9	43.4	**79.6**	56.2	**16.0**	**26.0**	**19.8**	33.2	83.3	47.5
+ AugMix	**81.5**	55.0	65.0	44.2	65.4	52.7	14.7	20.6	17.2	42.0	80.1	55.1
+ ANT	80.6	**59.5**	**75.6**	**50.0**	69.0	**57.9**	15.8	23.2	18.8	**43.3**	**89.0**	**58.2**
LFGAA+Hybrid	**71.6**	**57.9**	**70.4**	25.9	**81.8**	39.4	14.0	**39.7**	20.8	15.2	88.4	25.9
+ AugMix	66.1	56.3	64.6	20.9	78.5	33.0	13.4	**39.7**	20.0	**18.6**	91.3	**31.0**
+ ANT	70.7	56.6	68.8	**28.0**	80.4	**41.6**	**15.4**	38.8	**22.1**	16.4	**92.5**	27.8

Table 4.2 Results of LFGAA [1] trained with *AugMix* and *ANT*.

performs worse; in ZSL it degrades the results, except some cases (i.e. noise in SUN). In the *GZSL* setup, the results look better for AWA2 but for other datasets the degradation trend is still there.

ANT, for *LFGAA+SA* (see Figure 4.6), introduces better results compared to those of AugMix. In ZSL, except some isolated cases on SUN, consistent improvements are shown. In the *GZSL* setting the trend is the same. The recoveries in unseen classes are visibly better than seen classes. For *LFGAA+Hybrid* (see Figure 4.7), ANT fails to deliver. In ZSL, it introduces diverse affects but still does a better job than AugMix. In the *GZSL* setting, except AWA2 and some other cases (i.e. noise in CUB and SUN), it does introduce accuracy improvements. In overall, ANT does a visibly-better job, but can not deliver a tangible improvement in *LFGAA+Hybrid*.

4.4.5. Analysing corruption robustness

Our corruption results show many similarities with our observations on adversarial attacks presented in chapter 3., in terms of unseen/seen and ZSL/GZSL discrepancy, and adverse effect of defenses. We perform a similar set of analyses to see the characteristics of ZSL models under the impact of corruptions. The analyses presented here are primarily for the ALE model, but note that the insights apply to LFGAA as well.

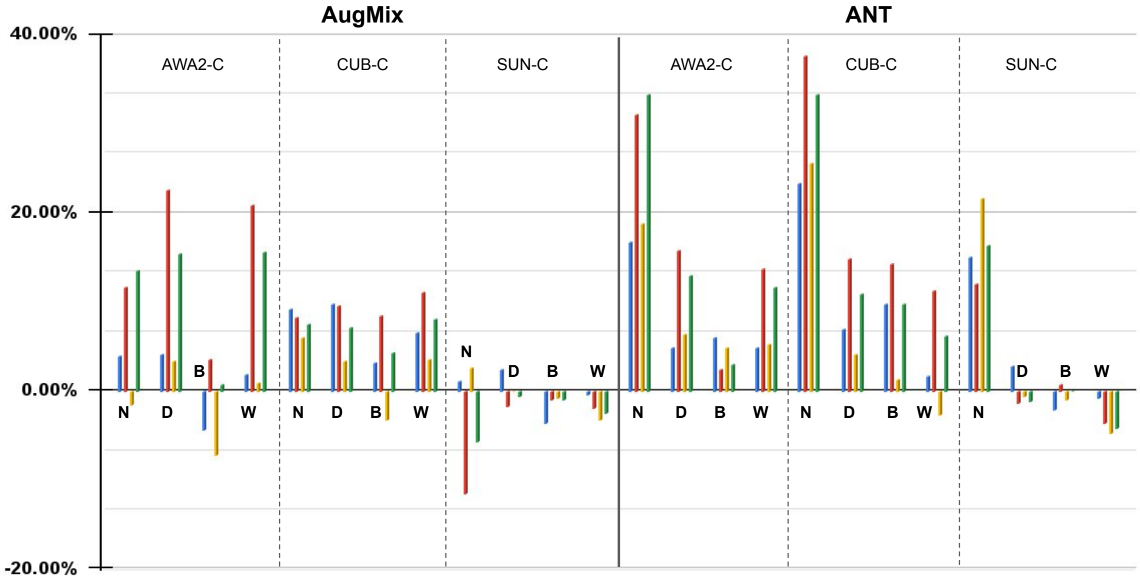


Figure 4.6 AugMix and ANT performance, as a % of clean results. LFGAA SA.

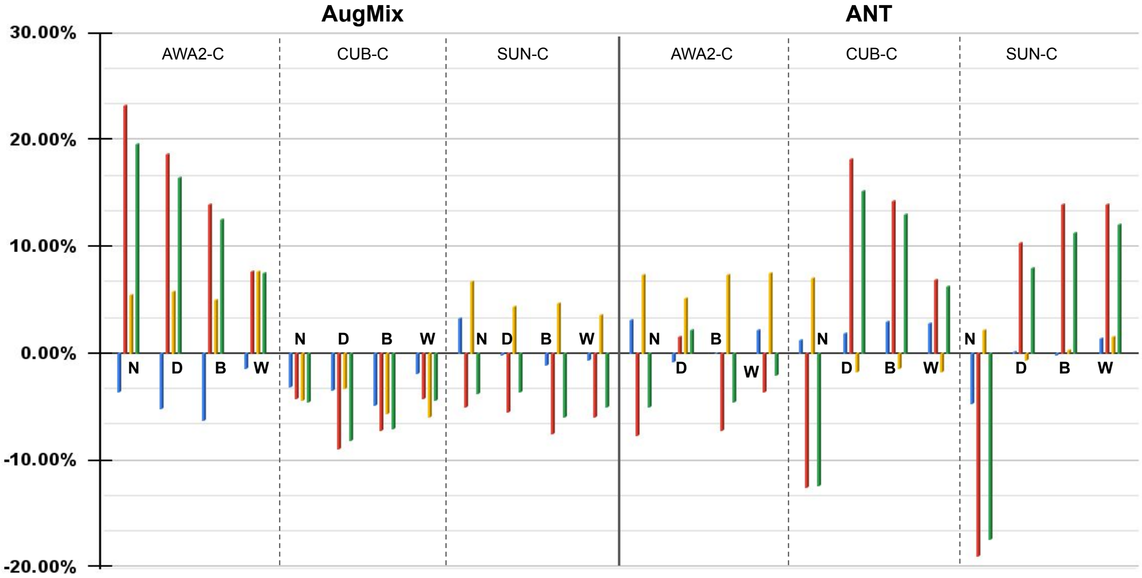


Figure 4.7 AugMix and ANT performance, as a % of clean results. LFGAA H.

Class transitions: False/Correct. We focus on the discrepancy between seen/unseen classes. Figure 4.2 shows that on AWA2, unseen classes are affected more severely but

Generalized Zero Shot

	C						S						A					
Class Type	U			S			U			S			U			S		
Transitions	CF	FC	FF	CF	FC	FF	CF	FC	FF	CF	FC	FF	CF	FC	FF	CF	FC	FF
Noise	79	4	78	73	7	79	81	3	85	81	3	84	75	2	61	38	18	47
Digital	59	6	68	48	12	59	60	4	67	58	5	67	64	3	50	30	16	39
Weather	61	5	70	53	10	62	63	4	71	61	4	69	60	3	51	33	16	42
Blur	63	5	72	55	12	64	83	3	79	71	4	78	79	3	68	54	14	55

Table 4.3 Categorization of prediction changes induced by corruption categories.

on SUN and CUB, both are affected pretty much the same. Following the practice of chapter 3., we investigate this behaviour by analyzing the class transitions for each corruption category. We calculate the following; out of all (originally) correctly predicted samples, what percentage have transitioned to false and out of all (originally) falsely predicted samples, what percentage have transitioned to correct to other false classes? The results are shown in Table 4.3. In Table 4.3, U and S columns are results for unseen and seen classes, respectively. CF, FC and FF are *correct-to-false* (as the percentage of all originally correct predictions), *false-to-correct* and *false-to-other-false* (as the percentage of all originally incorrect predictions) changes in %, represented as ratio averages.

We observe that the CF transitions tightly correlate with the results of Figure 4.2; stronger categories lead to higher CF transitions. Furthermore, the stronger categories lead to higher FF transitions, which is yet another indication of stronger corruptions leading to more class transitions. FC transitions are higher for *digital* corruptions (up to 16%), which has the lowest CF transition (*i.e.* 30% on AWA2 for seen classes), which partially explains why it failed to create a strong negative impact.

The comparison of seen and unseen class accuracy values provides key insights. First, CF transitions are higher for unseen classes, especially on AWA2 where unseen CF ratios basically double the seen CF ratios. Moreover, unseen FC transitions are visibly lower compared to seen classes, except on SUN where they are pretty similar. High CF and low FC indicates that unseen classes are affected disproportionately. Note that this was also the case for adversarial attacks, although the absolute accuracy values in chapter 3. showed otherwise for adversaries due to the *pseudo-robustness effect*. A similar situation occurs here too; Figure 4.2 shows unseen/seen discrepancy only for AWA2. On CUB and SUN, the discrepancy is not as clear. Unlike the adversarial scenario, however, the level of discrepancy between seen/unseen classes correlate well with the absolute reduction results of Table 4.3; on CUB and SUN, CF/FC difference between unseen/seen classes is not prominent therefore seen/unseen absolute accuracy values are close, whereas CF/FC difference between unseen/seen classes is quite prominent and seen/unseen absolute accuracy values are quite different. Finally, we see high FC values for seen classes; this means that the correct prediction probably has a high confidence, but not high enough to be correct prediction (*i.e.* model predicts incorrectly, but correct label has high confidence), therefore any *nudge* enforced to the image in the manifold is likely to lead to boundary transition towards the correct class, resulting into high FC transitions for seen classes.

Class transitions: Seen/Unseen. Here, we focus on the transitions from a seen/unseen class aspect. Following the practice of chapter 3., we calculate the following for all samples and average it for seen and unseen classes; out of *all* changed samples, what percent went to a seen or an unseen class? The results are given in Table 4.4. In Table 4.4, UU, US, SU and SS are unseen-to-unseen, unseen-to-seen, seen-to-unseen and seen-to-seen transitions, respectively.

The comparison of different categories shows minimal differences, hinting that the type of corruption has little to no role in the transition trends. We also see that originally seen classes are more likely to transition to another seen class, compared to originally unseen classes. Independent of the dataset and the corruption, class transitions happen decisively towards the seen classes. These trends are reminiscent of the findings of chapter 3. and we believe the core reasons are the same; the discrepancy between number of seen/unseen classes in the datasets, as well as the bias of the model towards seen classes, play a significant part in shaping the nature of the transitions.

A large part of the adversarial robustness analyses in chapter 3. are spearheaded by the fact

Generalized Zero Shot

	C				S				A			
Class Transition	UU	US	SU	SS	UU	US	SU	SS	UU	US	SU	SS
$Noise$	26	74	23	77	14	86	12	88	8	92	4	96
$Digital$	23	77	17	83	16	84	12	88	12	88	5	95
$Weather$	25	75	20	80	16	84	10	90	12	88	5	95
$Blur$	22	78	17	83	14	86	10	90	10	90	7	93

Table 4.4 Class transition averages (in %) for corruption categories.

that the ALE model is *weak*; it is not that accurate. When exposed to perturbations, there are common occurrences of attacked test sets having higher accuracy values. For corruptions, this happens just once; the brightness corruption shows less than 0.1% increase in ZSL accuracy on AWA2. Since this is clearly a negligible increase for an outlier (1 out of 75 corruptions), it is sensible to declare that the weakness of the model does not play a part against corruptions.

Adverse effect of defenses. As shown in Figures 4.3 to 4.4, a large part of the defense mechanisms fail to work. Their introduction results into worsening of the results, not just into *failure to recover*. In chapter 3., the results show the degrading effects of defenses (under adversarial attacks) had high correlation with *FCF* cases (*i.e.* original false prediction corrected by the attack, and then recovered to original prediction by the defense). In contrast, here we see that defenses do not work, because they simply *do not work*. Since there are practically no cases where a corruption increases the accuracy levels, we observe that *FCF* scenarios simply do not exist, therefore we believe that the negative effect of defenses are not tied to the *weakness* of the model.

Corrupting only the correct predictions. We also corrupt only the originally correct predictions and then defend them; our agenda is to potentially decouple the effects of model maturity and ZSL-specific behaviour. The results support our previous claim; the weakness of the model does not play a part. We see the same behaviour in all results; defenses which did not work *still* do not work. The isolated, single case of corruptions increasing the accuracy is eliminated, but this virtually has no effect on the overall trend. All the insights explained in Section 4.4.3. hold; ZSL/GZSL, unseen/seen discrepancies and corruption/defense performances are the same when we only corrupt the correct predictions.

ZSL vs GZSL. The model performance in ZSL and GZSL settings show that harmonic scores are consistently more severely affected than ZSL accuracy levels. This can be linked to the very nature of both settings; in ZSL, all test classes are balanced in the sense that *none* of them are seen during the training. In GZSL, class imbalance shows its effect and severely degrades unseen classes, which leads to poor harmonic scores. For defenses that do not work, there is no definitive answer as to if ZSL or GZSL scores are recovered in a better fashion. For defenses that work, GZSL accuracy levels are recovered whereas ZSL accuracies fail to do so. We credit this to several factors; first, GZSL methods are severely impacted already, so they have more *space* for recovery. Second, working defense methods produce models with higher GZSL accuracies, which might be resulting into better robustness.

Dataset characteristics. SUN and CUB both have high number of classes with few samples per class. AWA2, in contrast, has few classes with high per-class sample count. We observe the effects of this discrepancy in our results; AWA2 is the least affected dataset, especially in ZSL (*i.e.* reduction of 23% for digital category). Seen/unseen class discrepancy is most prominent on AWA2, probably due to the class imbalance further exacerbated with high per-class sample count. This is also reflected by quite low CF transitions. Highest FC transitions occur for AWA2 as well, hinting that AWA2 trained model outputs multiple confident predictions. Furthermore, highest number of transitions to seen classes also signal the bias of AWA2 trained models. Despite the fact that AWA2 is the least affected one under corruptions, it experiences the highest recovery rates when defended (*i.e.* 50% recovery in noise category). When defended with preprocessing defenses (*i.e.* spatial smoothing and TVM), we observe SUN getting the best returns, followed by CUB. Apart from that, SUN and CUB trained models share exhibit similar behaviour.

4.4.6. Comparing LFGAA and ALE

ALE vs LFGAA. The results of Figures 4.2 and 4.5 show that LFGAA is visibly more robust than ALE, especially with the inductive variant *LFGAA+Hybrid*. We credit the improved robustness of LFGAA to the previously mentioned factors that makes LFGAA a more powerful model in the first place.

Transductive vs Inductive. The results of Figure 4.5 show that *LFGAA+Hybrid* does a better job then *LFGAA+SA*. This is surprising, because *LFGAA+SA* leverages more data, which is often shown to improve robustness [189]. However, we hypothesize that the corruptions are potentially harming the self-adaptation process, which results into less effective prototypes.

Defense Performance for ALE and LFGAA. When we compare Figures 4.4, 4.6 and 4.7, we see that the defenses work approximately the same for ALE and *LFGAA+SA*. There are slight differences in performances across different datasets, but in average AugMix and ANT provide the most significant improvements. When we compare *LFGAA+Hybrid* and ALE, we see that ALE enjoys better defense performance.

Comparing the trends for ALE and LFGAA. We see that the trends presented for ALE in Section 4.4.5. apply to LFGAA as well; ZSL results are less affected than GZSL, unseen classes go through a more severe degradation and AWA2-trained models are the most robust ones.

4.5. Comparing adversarial and corruption robustness

We now combine our observations with our prior results on adversarial robustness from chapter 3. to conclude our analyses.

Model strength. The original performance of ALE does have an impact on adversaries, whereas it does not have an effect on corruptions. Adversarial attacks try to cross decision boundaries, often in an *efficient* manner (*i.e.* imperceptible, minimum perturbation) whereas corruptions do not focus on being efficient, as can be seen from the fact that Table 4.3 has lower FC than the corresponding results of chapter 3.. Adversaries often cross to the closest decision boundary, which in some cases result into *correcting* an originally incorrect prediction (*i.e.* upward accuracy spikes). Defenses also recover the originally false predictions (*i.e.* adverse effects of defenses). We see none of these trends for corruptions.

Unseen/seen discrepancy. We observe that unseen classes are affected more severely. This trend is apparent against both corruption and adversaries, hinting that the class imbalance affects them the same. However, due to model weakness, this effect is effectively *masked* for adversarial attacks, *i.e.* unseen classes go through less degradation than seen classes, which we call the *pseudo-robustness effect*. This effect is not observed for corruptions.

GZSL and ZSL. In corruption experiments, GZSL is affected more severely whereas for adversarial attacks, there is no definitive answer. In both cases, working defenses improve GZSL performance more so than they improve ZSL.

Datasets. Models trained on AWA2 are the most robust ones, owing to their high per-class sample count and potentially to their comparably fewer number of classes. The robustness difference between models trained on AWA2 and other datasets is smaller against corruptions, suggesting that simply shoving more data in the mix may not solve corruption robustness.

Defenses. THe adversarial defenses tested in chapter 3. do not work against corruptions. Although they work arguably well against adversarial attacks, they fail to deliver against corruptions (except some noise and blur types). Despite the careful design of adversaries, this highlights that their effect on images are somewhat limited; corruptions cover a significantly wider range of effects and probably require an even *more* fundamental solution.

4.6. Conclusion

The robustness of ML models has attracted significant attention in the recent years, however it has primarily been approached from a supervised perspective. In this study, we present the first comprehensive study to analyse the robustness of discriminative ZSL models against common image corruptions. We subject several commonly used discriminative ZSL models to corruptions and corruption defenses, and also create multiple corruption benchmark datasets for ZSL. Our baseline results show that the discriminative ZSL models are not robust at all, due to the severe class imbalance and model weakness. Our results further indicate that although some defense methods do work, they fail to do so in a tangible manner, underlining the necessity of further research in the topic. We also show that despite the failure of defense methods in improving robustness, they help set new high accuracy levels for our ZSL models. We emphasize the important differences in robustness between seen/unseen classes and ZSL/GZSL settings. Finally, we conclude by jointly analysing and comparing the results on corruptions and adversaries, and provide a bigger picture of discriminative ZSL robustness.

5. HybridAugment++: Unified Frequency Spectra Perturbations for Model Robustness

In this chapter, we focus on designing methods aimed at improving the robustness of models to distribution shifts. We adopt a frequency-centric approach and propose not one but two methods; *HybridAugment* and *HybridAugment++*. By performing a hierarchical data augmentation in frequency spectra, our method improves clean and robustness performance on multiple datasets and multiple distribution shifts.

5.1. Introduction

The last decade witnessed machine learning (ML) elevating many methods to new heights in various fields. Despite surpassing human performance in multiple tasks, the *generalization* of these models are hampered by distribution shifts, such as adversarial examples [30], common image corruptions [36] and out-of-distribution samples [191]. Addressing these issues are of paramount importance to facilitate the wide-spread adoption of ML models in practical deployment, especially in safety-critical ones [192, 193], where such distribution shifts are simply inevitable.

Distribution shift-induced performance drops signal a gap between how ML models and us humans perform perception. Several studies attempted to bridge, or at least understand, this gap from architecture [149, 151, 152] and training data [10, 36, 138–142] centric perspectives. An interesting perspective is built on the frequency spectra of the training data; convolutional neural networks (CNN) are shown to leverage high-frequency components that are invisible to humans [39] and also shown to be reliant on the amplitude component, as opposed to the phase component us humans favour [10]. Several studies leveraged frequency spectra insights to improve model robustness. These methods, however, either leverage cumbersome ensemble models [152], formulate complex augmentation regimes [157, 158] or focus on a single robustness venue [157, 158, 194] rather than improving the broader

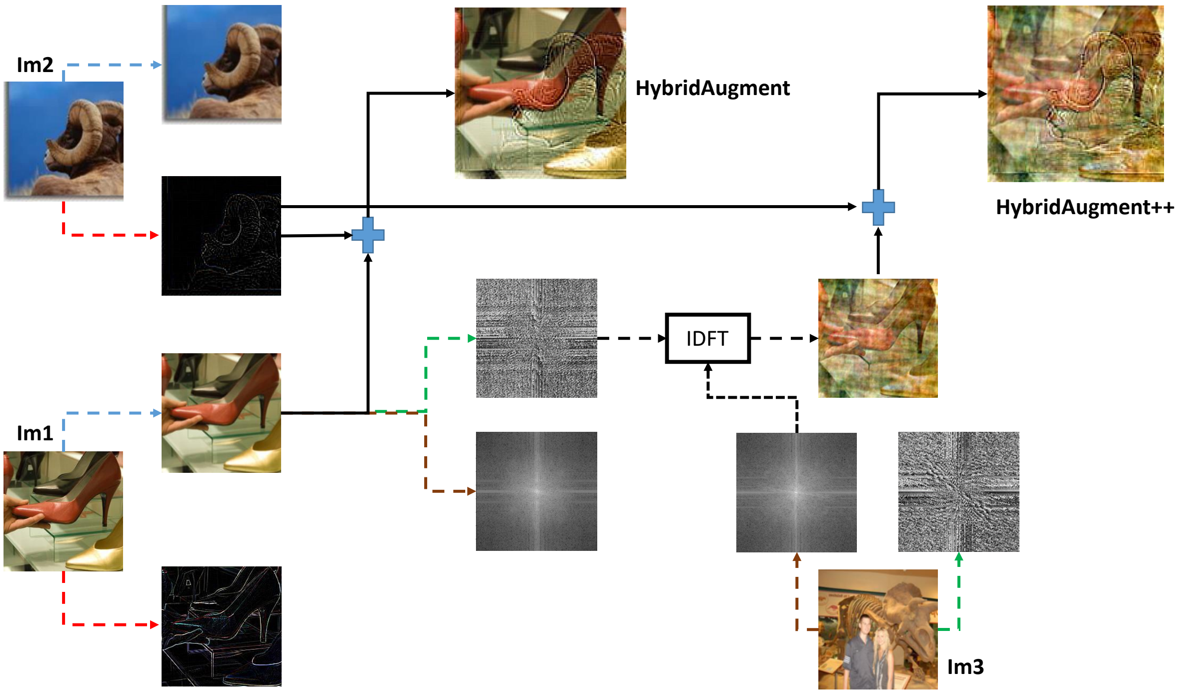


Figure 5.1 Our methods. Colors: Phase, amplitude, low & high frequency.

robustness to various distribution shifts. Furthermore, it is imperative to preserve, if not improve, the clean accuracy levels of the model while improving its robustness.

Our work aims to improve the robustness of CNNs to various distribution shifts. Inspired by the frequency spectra based data augmentations, we propose *HybridAugment* which is based on the well-known hybrid images [40]. Based on the observation that the label information of images are predominantly related to the low-frequency components [195, 196], *HybridAugment* simply swaps high-frequency and low-frequency components of randomly selected images in a batch, regardless of their class labels. This forces the network to focus on the low-frequency information of images and makes the models less reliant on the high-frequency information, which are often shown to be the root cause of robustness issues. *HybridAugment* is implemented in three lines of code, produces virtually no training overhead and improves both adversarial and corruption robustness while preserving or improving the clean accuracy.

Additionally, we set our eyes on jointly exploiting the contributions of frequency spectra augmentation methods while unifying them into a simpler, single augmentation regime. We then propose *HybridAugment++*, which performs hierarchical perturbations in the frequency spectra. Exploiting the fact that the phase component carries most of the information in an image [10], *HybridAugment++* first decomposes images into high and low-frequency components, swaps the amplitude and phase of the low frequency component with another image, and then combines this augmented low-frequency information with the high-frequency component of a random image. *HybridAugment++* forces the models to rely on the phase and the low-frequency information, which helps align them with human perception. As a result, *HybridAugment++* further improves adversarial and corruption robustness, while further improving the clean accuracy against several alternatives. See Figure 5.1 for a diagram of our proposed methods.

Our main contributions can be summarized as follows.

- We propose *HybridAugment*, a simple data augmentation method that helps models rely on low-frequency components of data samples. It is implemented in just three lines of code and has virtually no overhead.
- We extend *HybridAugment* and propose *HybridAugment++*, which performs hierarchical augmentations in the frequency spectra to help models rely on low-frequency and phase components of images.
- *HybridAugment* improves adversarial and corruption robustness of multiple CNN models, while preserving (or improving) the clean accuracy. *HybridAugment++* outperforms many alternatives by further improving corruption, adversarial and clean accuracies on multiple benchmark datasets.

5.2. Related Work

Robust Generalization - Adversarial. Adversarial ML has been studied intensively since their discovery [30], resulting into numerous attack [30, 96, 103] and defense [120, 121, 125, 197] methods borne out of an arms race that is still very much active. Notable attacks include

FGSM [96], DeepFool [103], C&W [99] where AutoAttack [2] is now a widely used attack for adversarial evaluation. The defense methods either diversify the training distribution with attacked images [197, 198], purify the adversarial examples [125, 127] or detect whether an image is adversary or not [120, 124].

Robust Generalization - Corruptions. Common image corruptions might have various causes, and they occur more frequently than adversaries in practice. Numerous datasets simulating these effects have been released to facilitate standard evaluations [36, 132, 135]. The methods addressing corruption robustness can be largely divided into two; architecture-centric and data-centric methods. Architecture-centric methods include neural architecture search for robust architectures [199], focusing on subnets [153], rectifying batch normalization [149], wavelet based layers [200] and forming ensembles [151, 152]. The data-centric methods are arguably more prominent in the literature; adversarial training [144, 197], cascade augmentations [36, 138], augmentation networks [142, 145], learned augmentation policies [201] , shape-bias injection [146, 147], style augmentation [8], using fractals [141], soft-edge driven image blending [139] and max-entropy image transformations [140] are all shown to improve corruption robustness at varying degrees.

Robust Generalization - Frequency Aspect. The generalization of neural networks have been analysed extensively. Specifically, several frequency-centric studies showed that CNNs tend to rely on the high-frequency information us humans can not see [39], or rely more on the amplitude component than phase component us humans tend to favour [10]. Similarly, when models are trained on low-pass or high-pass filtered images, models trained on high-pass filtered images have significantly higher accuracy than the models trained on low-pass filtered images, although high-pass filtered images look like random noise to us humans [201]. Multiple studies confirm that models reliant on low-frequency components are more robust [195, 196]. Interestingly, frequency analyses present a different interpretation of the robustness-accuracy trade-off; many methods that improve clean accuracy force networks to rely on the high-frequency components, which might sacrifice robustness [39].

Robust Generalization - Frequency-Centric Methods. A trade-off in frequency-based data augmentations is that one should not sacrifice the other; training on high-frequency augmentations can improve robustness to high-frequency corruptions, but tend to sacrifice the low-frequency corruption robustness or the clean accuracy [152, 201, 202]. Frequency-centric methods include biasing Jacobians [202], swapping phase and amplitude of random images [10], perturbing phase and amplitude spectra along with consistency regularization [157], frequency-band expert ensembles [152], frequency-component swapping of same-class samples [203] and wavelet-denoising layers [200]. Note that there is a considerable literature on frequency-centric adversarial attacks, but since we primarily focus on methods to improve robustness, they are not mentioned here.

A concurrent work is [203], where hybrid-image based augmentation is also used. We have, however, several key advantages; we i) lift the restriction of sampling from the same classes for augmentation, leading to a more diverse training distribution, ii) also present *HybridAugment++* that performs phase/amplitude swap specifically in low-frequency components, iii) report improvements on corruption and adversarial robustness, as well as clean accuracy on multiple benchmark datasets (CIFAR-10/100, ImageNet). Note that other frequency-based methods either train with ImageNet-C corruptions [152], report only corruption results [157], rely on external data [141] or use external models [142]. *HybridAugment* is simple to implement, can be easily plugged in to existing training pipelines and present a broader analyses of various robustness venues.

5.3. Method

In this section, we formally define the problem, motivate our work and then present our proposed techniques.

5.3.1. Preliminaries

Let $F(x; W)$ be an image classification CNN trained on the training set $TR_x = (x_i, y_i)_{i=1}^N$ with N samples, where x and y correspond to images and labels. The clean accuracy (CA) of

$F(x;W)$ is formally defined as its accuracy over a clean test set $TE_x = (x_j, y_j)_{j=1}^{M}$. Assume two operators $A(\cdot)$ and $C(c, s)$ that adversarially attacks or corrupts TE_x with corruption category c and severity s. Let ATE_x and CTE_x be adversarially attacked and corrupted versions of TE_x, and let $F(x;W)$ have a robust accuracy (RA) on ATE_x and a corruption accuracy (CRA) on CTE_x. A robust $F(x;W)$ has minimal gaps between CA and RA, and CA and CRA. Ideally, CA is preserved or improved in the robustification process.

What we know. We first revisit key points in frequency-centric analyses; i) CNNs favour high-frequency content [39], ii) adversaries and corruptions primarily operate in high-frequency [195], but some also operate in low-frequency [36], iii) images are dominated by low-frequency information [152] and iv) models relying on low-frequency components are more robust [195, 196]. The trade-off presents itself naturally; low-frequency reliant models are more robust but might miss out on clean accuracy brought by the high-frequency components.

5.3.2. HybridAugment

We hypothesize that a sweet spot in the robustness/accuracy trade-off can still be found; unlike the *hard* approaches that completely rule out the reliance on high-frequency components, we propose to *reduce* the reliance on them. To this end, we adopt a data augmentation approach that aims to diversify TR_x by an operation $HA(\cdot)$. Keeping the strong relation intact between labels and low-frequency content (i.e. labels comes from low-frequency-component image), we propose to swap high and low-frequency components of images in a batch on-the-fly. Unlike [203], we *do not* restrict the images to belong to the same class; this diversifies the training distribution even further while preserving the image semantics. We call this *HybridAugment*, and it is defined as

$$HA_P(x_i, x_j) = LF(x_i) + HF(x_j) \tag{22}$$

where x_i is the input image and x_j is a randomly sampled image in the same batch as x_i. *HF* and *LF* are high and low-frequency components of images x_i and x_j, and they are defined as

$$\begin{aligned} LF(x) &= GaussBlur(x) \\ HF(x) &= x - LF(x) \end{aligned} \tag{23}$$

where $GaussBlur$ results into a low-frequency image. Note that a similar outcome is possible by using Discrete Fourier Transforms (DFT), swapping the frequency bands and then applying Inverse DFT (IDFT). We find the gaussian blur operation to be faster and better in practice.

Following the practice of [10], we propose two variants of *HybridAugment*; one that operates over multiple images (*Hybrid-P*) and one that operates on a single image (*Hybrid-S*). Essentially, the multiple image version is Equation 22, where two images are sampled to create an augmented image. In the single image variant, x_j is a different view of x_i created by applying N randomly sampled augmentation operations. The single image variant is defined as

$$HA_S(x_i) = LF(Aug(x_i)) + HF(\hat{Aug}(x_i)) \tag{24}$$

where Aug and $\hat{Aug}$ correspond to two sets of randomly sampled augmentation operations. Note that multiple and single versions can work in tandem (*Hybrid-PS*), and actually outperform single and multiple image versions.

5.3.3. HybridAugment++

The frequency analysis is a vast literature, however, two core aspects often stand out; frequency-bands (i.e. low, high) and the decomposition of signals into amplitude and phase. *HybridAugment* covers the former and tightly relates them to robustness and model

```
def hybrid_augment_paired(x_batch, prob, blur_fnc, is_ha_plus):
    #x_batch: batch of training images
    #prob: probability value [0,1]
    #blur_fnc: blurring function
    #is_ha_plus: True for HA++, false for HA

    p = random.uniform(0,1)
    if p > prob:
        return x

    batch_size = x_batch.size()[0]
    index = torch.randperm(batch_size)

    lfc = blur_fnc(x_batch)
    hfc = x - lfc
    hfc_mix = hfc[index]

    if is_ha_plus:
        #Based on the APR method.
        p = random.uniform(0,1)
        if p > 0.6:
            lfc  = lfc
        else:
            index_p = torch.randperm(batch_size)
            phase1, amp1 = fft(lfc)
            lfc_mix = lfc[index_p]
            phase2, amp2 = fft(lfc_mix)
            lfc = ifft(phase1, amp2)

    hybrid_ims = lfc + hfc_mix
    return hybrid_ims
```

Figure 5.2 Pseudocode of *HybridAugment* and *HybridAugment++* (paired).

accuracy, and shows competitive results in various robustness benchmarks (see Section 5.4.). The latter is investigated in APR [10], where phase is shown to be the more relevant component for correct classification, and training models based on their phase labels and swapping amplitude components of images randomly lead to more robust models. Note that frequency-bands and phases/amplitude discussions are orthogonal, since phase does not correspond to low-frequency and amplitude does not correspond to high-frequency. The phase component is shown to include more spatial information [10], which makes sense since amplitude is just a global representation (without spatiality) based on different frequency bases.

We hypothesize that these two approaches can be complementary; a model more reliant on low-frequency components and spatial information (i.e. phase) can further improve model robustness. Inspired by the successes of cascaded augmentation methods [36, 138, 142], we unify the two core aspects of the frequency spectra into a single, hierarchical augmentation method *HybridAugment++*. *HybridAugment++* is defined as

$$HA_P^{++}(x_i, x_j, x_z) = APR_P(LF(x_i), x_z) + HF(x_j) \tag{25}$$

where APR_P is defined as [10]

$$APR_P(x_i, x_z) = IDFT(A_{x_z} \otimes e^{i.P_{x_i}}) \tag{26}$$

where $\otimes$ is element-wise multiplication, A is the amplitude and P is the phase component. Similar to *HybridAugment* and APR, we also define a single-image version of *HybridAugment++* as

$$HA_S^{++}(x_i) = APR_S(LF(Aug(x_i))) + HF(\hat{Aug}(x_i)) \tag{27}$$

where APR_S is defined as [10]

$$APR_S(x_i) = IDFT\left(A_{\bar{Aug}(x_i)} \otimes e^{i.P_{\overline{Aug}(x_i)}}\right) \tag{28}$$

where Aug, $\hat{Aug}$, $\bar{Aug}$ and $\overline{Aug}$ are different sets of randomly sampled augmentation operations. Note that *HybridAugment++* is an essentially a framework; one can use different single and paired image augmentations (phase/amplitude or frequency-bands), either individually or together, and can still achieve competitive results (see Section 5.4.). Note that swapping phase/amplitudes first and then performing *HybridAugment* is an alternative, but it exhibits poor performance in practice; dividing the phase component into frequency-bands is not an interpretable operation as frequencies of the phase component are not well defined. See Figure 5.2 for the pseudo-code of our methods.

5.4. Experimental Results

We evaluate the effectiveness of *HybridAugment* and *HybridAugment++* in three distribution shifts; adversarial attacks, common image corruptions and out-of-distribution detection.

5.4.1. Datasets and Evaluation Metrics

Datasets. We use CIFAR-10, CIFAR-100 [204] and ImageNet [205] for training our models. Both CIFAR datasets are formed of 50.000 training images with a size of 32×32. ImageNet dataset contains around 1.2 million images of 1000 different classes. Corruption robustness evaluation is performed using the corrupted versions of the datasets; CIFAR-10-C, CIFAR-100-C and ImageNet-C datasets [9] are corrupted versions of the respective datasets' test sets. For each dataset, corruptions are simulated for 4 categories (noise, blur, weather, digital) with 15 different corruption types, each with 5 severity levels. For adversarial robustness, we use AutoAttack [2] on CIFAR-10 test set. Out-of-distribution detection is evaluated on SVHN [206], LSUN [207], ImageNet and CIFAR-100, and their fixed versions [208].

Evaluation metrics. We report top-1 accuracy as the clean accuracy. Adversarial robustness is evaluated with robust accuracy, which is the top-1 accuracy on adversarially attacked test sets. Corruption robustness is evaluated with Corruption Error (CE) $CE = \sum_1^5 E_{c,s}^F / \sum_1^5 E_{c,s}^{AlexNet}$. CE calculates the average error of the model F on a corruption type, normalized by the corruption error of AlexNet [23]. CE is calculated for all 15 corruption types, and their average Mean Corruption Error (mCE) is used as the final robustness metric. Out-of-distribution detection is evaluated using the Area Under the Receiver Operating Characteristic Curve (AUROC) metric [209].

5.4.2. Implementation Details

Architectures. We choose architectures commonly used in the robustness literature for a fair comparison [10]; ResNeXT [5], All-Convolutional Network [56], DenseNet [6], WideResNet and ResNet18 [4] are used in CIFAR-10 and CIFAR-100 experiments, whereas ResNet50 is used for ImageNet experiments.

Implementation Details. For CIFAR experiments, all architectures are trained for 200 epochs with SGD, where an initial learning rate of 0.1 decays after every 60 epochs. We use the last checkpoints for evaluation and do not perform any explicit hyperparameter tuning. Paired and single variants of *HybridAugment* and *HybridAugment*++ are applied in each iteration with probabilities 0.6 and 0.5, respectively. Standard training augmentations are random horizontal flips and cropping. When a singles augmentation is used, the input image is augmented with Aug randomly sampled among [*rasterize, autocontrast, equalize, rotate, solarize, shear, translate*]. Note that these do not overlap with test corruptions. For ImageNet experiments, we train for 100 epochs with SGD, where an initial learning rate of 0.1 is decayed every 30 epochs. Data augmentations and their probabilities are the same as above.

We note that we use the same checkpoints for all evaluations; we do not train separate models for corruption and out-of-distribution detection. In adversarial analysis, for a fair comparison with [10], we train our model with our augmentations and FGSM-based adversarial training. Finally, we note that we use the labels of the low-frequency image as the ground-truth labels. We have tried using the high-frequency image labels instead, but this leads to severe degradation in overall performance. All models are trained with standard cross-entropy loss, where both original and augmented batches are used to calculate the loss.

5.4.3. Corruption Robustness Results

CIFAR-100/100. We first present our results on CIFAR-10 and CIFAR-100. We compare with various methods, such as CutOut [210], Mixup [211], CutMix [212], adversarial training (AT) [197], AutoAugment (AA) [38], AugMix [36] and APR [10]. The corruption results

	Orig	Cutout	Mixup	CutMix	AT	AugMix	AA	APR$_S$	HA$_S$	HA$_S^{++}$	APR$_P$	HA$_P$	HA$_P^{++}$	APR$_{PS}$	HA$_{PS}$	HA$_P^{++} + APR_S$	HA$_P^{++} + HA_S$	HA$_{PS}^{++}$
AllConv	30.8	32.9	24.6	31.3	28.1	15	29.2	14.8	16.8	**13.9**	21.5	20.8	**16.7**	11.5	12	10.9	10.9	**10.7**
DenseNet	30.7	32.1	24.6	33.5	27.6	12.7	26.6	12.3	15	**11.1**	20.3	18.4	**14.2**	10.3	10.9	10.1	10	**9.5**
WResNet	26.9	26.8	22.3	27.1	26.2	11.2	23.9	10.6	13.6	**10.0**	18.3	16.4	**13.2**	9.1	9.9	8.5	8.5	**8.3**
ResNeXt	27.5	28.9	22.6	29.5	27	10.9	24.2	11.0	13.2	**9.99**	18.5	17.6	**13.2**	9.1	10.3	8.3	8.2	**7.9**
ResNet18	-	-	-	-	-	-	-	9.9	12.2	**9.34**	17.0	18.3	**15.2**	9.1	9.3	8.6	8.4	**8.2**
Mean	29.0	30.2	23.5	30.3	27.2	12.5	26.0	11.7	14.1	**10.9**	19.1	18.3	**14.5**	9.8	10.4	9.2	9.2	**8.9**
AllConv	56.4	56.8	53.4	56	56	42.7	55.1	39.8	43.0	**38.9**	47.5	44.7	**41.7**	35.9	36.5	**34.4**	34.6	**34.4**
DenseNet	59.3	59.6	55.4	59.2	55.2	39.6	53.9	38.3	41.3	**37.3**	49.8	45.6	**41.8**	35.8	36.1	34.3	34.3	**33.4**
WResNet	53.3	53.5	50.4	52.9	55.1	35.9	49.6	35.5	38.1	**33.9**	44.7	43.13	**39.3**	32.9	34.2	31.5	31.4	**31.2**
ResNeXt	53.4	54.6	51.4	54.1	54.4	34.9	51.3	33.7	35.6	**31.1**	44.2	41.2	**36.4**	31.0	31.5	30.5	29.0	**28.8**
ResNet18	-	-	-	-	-	-	-	33.0	35.6	**32.1**	49.2	45.5	**44.6**	31.8	31.8	30.3	30.4	**29.9**
Mean	55.6	56.1	52.6	55.5	55.2	38.3	52.5	36.0	38.7	**34.6**	47.0	44.0	**40.7**	33.4	34.0	32.2	31.9	**31.5**

Table 5.1 Corruption robustness mCE on CIFAR-10 (first 6 rows) and CIFAR-100.

	Orig	Cutout	Mixup	CutMix	AT	AugMix	AA	APR$_S$	HA$_S$	HA$_S^{++}$	APR$_P$	HA$_P$	HA$_P^{++}$	APR$_{PS}$	HA$_{PS}$	HA$_P^{++} + APR_S$	HA$_P^{++} + HA_S$	HA$_{PS}^{++}$
AllConv	93.9	93.9	93.7	93.6	81.1	93.5	93.5	93.5	94.1	93.9	**94.5**	93.9	94.0	94.3	**94.5**	**94.5**	94.4	94.3
DenseNet	94.2	**95.2**	94.5	94.7	82.1	95.1	95.2	94.9	94.7	95.0	95.0	93.1	93.2	**95.2**	94.9	94.8	95.0	94.8
WResNet	94.8	95.6	95.1	95.4	82.9	95.1	95.2	95.0	95.3	95.4	95.2	93.2	92.0	**95.7**	95.0	**95.7**	95.3	95.3
ResNeXt	95.7	95.6	95.8	96.1	84.6	95.8	96.2	95.5	95.3	95.7	95.5	93.5	92.9	**96.1**	95.2	95.6	96.0	95.9
ResNet18	92.2	-	-	-	-	-	-	**95.6**	95.5	**95.6**	94.9	90.9	89.7	95.0	95.4	95.0	95.1	95.0
Mean	94.2	95.0	94.7	94.9	82.6	94.8	95.0	94.9	94.9	95.1	95.0	92.9	92.3	**95.2**	95.0	95.1	**95.2**	95.1
AllConv	74.9	-	-	-	-	-	-	75.3	75.0	**75.8**	74.8	74.08	74.7	75.2	**75.8**	75.7	75.6	75.2
DenseNet	71.4	-	-	-	-	-	-	75.8	**76.0**	75.6	71.5	71.4	71.7	75.6	74.9	75.5	75.6	75.9
WResNet	72.1	-	-	-	-	-	-	76.2	**76.8**	76.2	70.4	71.3	71.7	**76.8**	74.8	76.1	76.3	76.0
ResNeXt	75.0	-	-	-	-	-	-	78.8	**79.4**	**79.4**	71.1	73.5	74.3	79.1	77.3	77.8	79.1	78.8
ResNet18	70.9	-	-	-	-	-	-	77.0	**77.4**	77.1	63.7	65.3	61.9	76.1	75.6	76.1	76.2	76.5
Mean	72.9	-	-	-	-	-	-	76.6	**76.9**	76.8	70.3	71.1	70.8	76.5	75.6	76.2	76.5	76.4

Table 5.2 Clean accuracy values on CIFAR-10 (first 6 rows) and CIFAR-100.

are shown in Table 5.1. Note that all results except ResNet18 are taken from [10] for a fair comparison; we also experiment with ResNet18, and when doing so, we train all APR variants ourselves using the authors' official codebase. Note that we do not train other methods with ResNet18 as the result trends are already clear with the other four architecture.

HybridAugment. First we focus on *HybridAugment*. In single augmentations (denoted with subscript s), it outperforms the original clean trained model and several other methods, but it lags behind APR. In paired augmentations, it outperforms its main competitor APR_P on all architectures and both datasets. Combining singles and augmentations further improves the corruption performance of all models; HA_{PS} outperforms all others and is competitive against APR_{PS}.

HybridAugment++. We see decisive improvements with *HybridAugment++*; HA_S^{++} already outperforms every other method except APR_{PS}. In pairs, HA_P^{++} is significantly better than APR_P and HA_P. When single and paired augmentations are combined, HA_{PS}^{++} significantly outperforms every other method on every possible architecture, with around 10% relative improvement over the next best method. These results verify our hypothesis that phase/amplitude and frequency-band centric frequency augmentations work together well, and they introduce significant improvements in robustness.

We report results with even more variants of our framework; we combine HA_P^{++} and APR_s, and also HA_P^{++} and HA_s. We note that these two variants actually outperform both HA_{PS} and APR_{PS} and are only dwarfed by HA_{PS}^{++}. The possibilities expand combinatorially, therefore we do not cover all of them, but these additional variants further support our hypothesis.

Clean Accuracy. The clean accuracy values of the same models reported in Table 5.1 are shown in Table 5.2. Note that the CIFAR-10 values (except ResNet18 since it is not reported) in Table 5.2 are taken from [10] for fair comparison. All the others are models trained by us since there is no pretrained model/reported result for the others. The results of Table 5.2 show us that both *HybridAugment* and *HybridAugment++* has achieved a good spot in robustness-accuracy trade-off; except one or two cases, both of them improve clean accuracy over the original models. This shows *HybridAugment* and *HybridAugment++* are valuable not only for corruption robustness, but also for clean accuracy improvements across multiple architectures and datasets.

Compared to other methods on CIFAR-10, $HA^{++} + HA_S$ essentially ties with APR_{PS}, showing the value of our methods. Note that our other variants are not far off either; HA_{PS}^{++} and HA_{PS} are 0.1 and 0.2 shy from the best mean accuracy, respectively. On CIFAR-100, HA_S takes the lead comfortably, whereas other variants of our method are either better or on-par with APR variants.

		Noise			Blur				Weather				Digital				
Method	Test Error	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Brightness	Contrast	Elastic	Pixel	JPEG	mCE
Standard	23.9	79	80	82	82	90	84	80	86	81	75	65	79	91	77	80	80.6
Patch Uniform	24.5	67	68	70	74	83	81	77	80	74	75	62	77	84	71	71	74.3
AA	22.8	69	68	72	77	83	80	81	79	75	64	56	70	88	57	71	72.7
Random AA	23.6	70	71	72	80	86	82	81	81	77	72	61	75	88	73	72	76.1
MaxBlur Pool	23	73	74	76	74	86	78	77	77	72	63	56	68	86	71	71	73.4
SIN	27.2	69	70	70	77	84	76	82	74	75	69	65	69	80	64	77	73.3
AugMix	22.4	65	66	67	70	80	66	66	**75**	72	67	58	58	79	69	69	68.4
APR$_S$	24.5	61	64	60	73	87	72	81	72	67	62	56	70	83	79	71	70.5
APR$_P$	24.4	64	68	68	70	89	69	81	69	69	55	57	58	85	66	72	69.3
APR$_{PS}$	24.4	55	61	54	68	84	68	80	62	62	49	53	57	83	70	69	65.0
APR$_{PS}*$	24.4	62	68	64	72	86	72	79	**66**	67	**51**	58	**61**	86	66	72	68.9
HA$_P^{++}$	23.5	64	66	67	71	88	72	78	70	69	59	58	64	84	**61**	69	69.7
HA$_{PS}$	**23.2**	66	67	62	72	85	77	**77**	77	71	65	58	69	**83**	63	69	71.2
HA$_{PS}^{++}$	23.7	**57**	**61**	**57**	**69**	**85**	**70**	78	67	**66**	58	**57**	63	85	63	**67**	**67.3**

Table 5.3 Clean accuracy and corruption robustness on ImageNet.

ImageNet. We now assess whether our methods can scale to ImageNet. We compare against various methods, such as SIN [145], PatchUniform, AutoAugment (AA), Random AA [38], MaxBlurPool and AugMix [36]. The results are shown in Table 5.3. Note that methods listed before APR_{PS} * are directly taken from [10]. Although the pretrained model for APR_{PS} is provided, since the authors do not provide an evaluation script, we wrote our own and there are discrepancies in the results. The results below and including $APR_{PS}*$ are evaluated with our script.

The results show that all of our variants produce higher clean accuracy compared to APR, showing the value of our method in improving model accuracy. *HybridAugment* results are competitive in corruption accuracy, but HybridAugment++ outperforms both APR and others in corruption accuracy, while being 0.5 shy of our best clean accuracy. Note that since APR results became worse with our script, it is likely that our results will be better if evaluated with their scripts. This issue can be remedied, but the trend remains the same; *HybridAugment++* outperforms others in corruption accuracy and significantly improves clean accuracy.

Qualitative results. We provide GradCam visualizations of *HybridAugment++* against various corruptions in Figure 5.3 with ImageNet validation images. We sample corruptions from each category; noise, motion blur, fog, pixelate and contrast corruptions are shown from top to bottom. In the first four rows, it is apparent that corruptions lead to the standard model focusing on the wrong areas, leading to misclassifications. Note that this is the case for APR as well; it can not withstand these corruptions whereas *HybridAugment* still focuses on where matters, and manages to predict correctly. The fifth row shows another failure

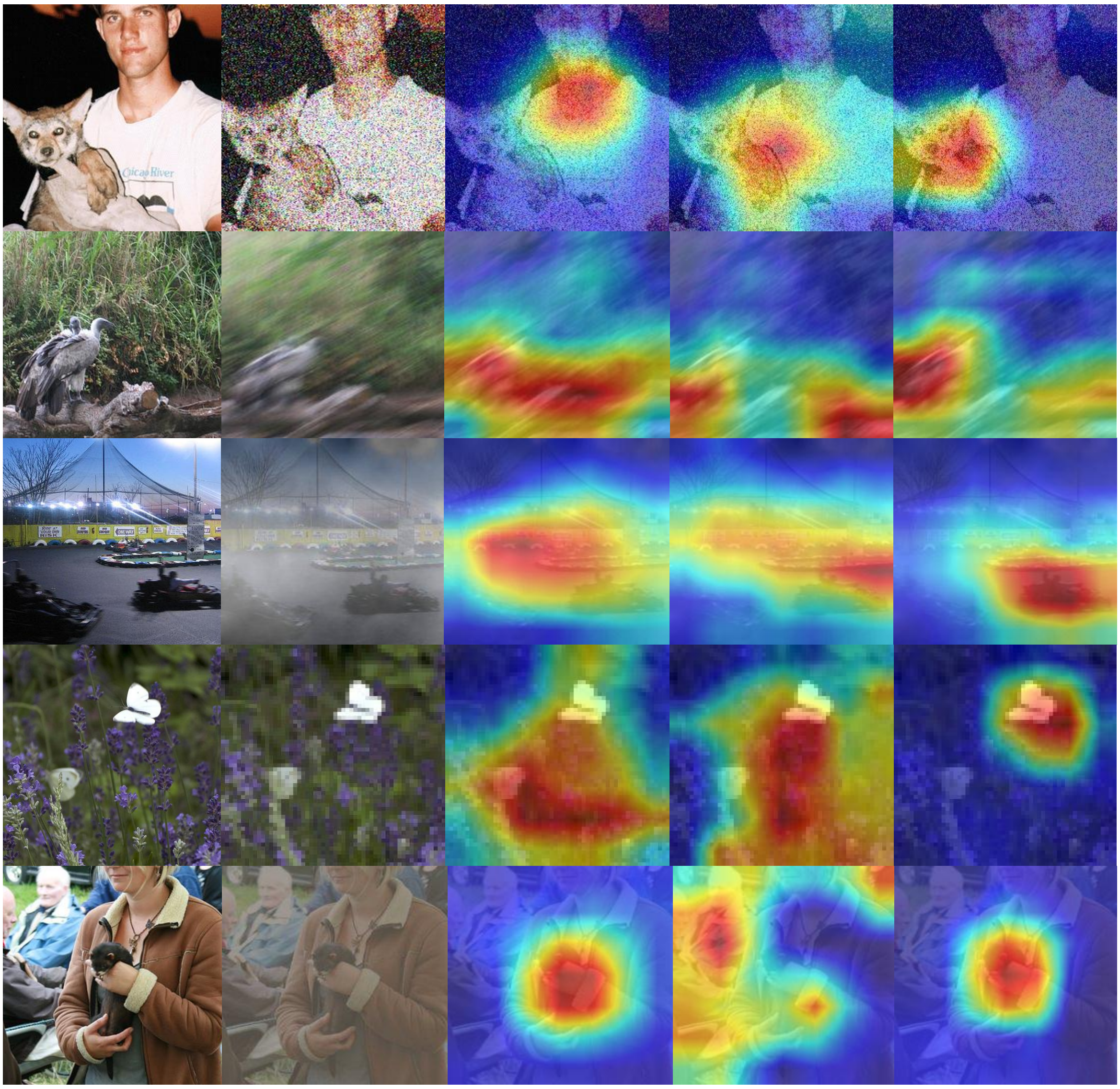


Figure 5.3 L-R: Input, corruption, baseline, APR [10] and our GradCAM [11].

mode; despite the corruption, standard model manages to predict correctly but APR loses its focus and leads to misprediction. *HybridAugment* does not break what works; this case visualizes the ability of *HybridAugment* to improve clean accuracy.

	AT	Cutout	APR$_P$	APR$_S$	APR$_{SP}$	APR$_{SP}$*	HA$_S$	HA$_S^{++}$	HA$_P$	HA$_P^{++}$	HA$_{PS}$	HA$_{PS}^{++}$
CA	83.3	81.3	85.3	83.5	84.3	84.4	**86.5**	85.0	85.5	85.4	85.0	82.8
RA	43.2	41.6	44.0	45.0	45.7	45.4	44.1	45.4	42.1	43.5	44.8	**46.0**

Table 5.4 Clean and robust accuracy on CIFAR-10 under AutoAttack [2].

5.4.4. Adversarial Robustness Results

We now present our results on adversarial robustness in Table 5.4. All results except ours are taken from [10]. Note that in the table, $AT + APR_{SP}$ and $AT + APR_{PS}$ * are the same experiments, except the one denoted with * is trained by us using the authors' code. We compare our results with APR, Cutout and FGSM-based adversarial training [197].

Our results show that there is no clear winner in all fronts; with HA_S we obtain the best clean accuracy and with with HA_{PS}^{++} we obtain the best robust accuracy. All our variants are better than the widely accepted adversarial training (AT) baseline in nearly all cases, which shows the effectiveness of our method. Our variants do quite well in clean accuracy and outperform others in nearly all cases. HA_S^{++} offers arguably the best trade-off; it ties with APR_{SP} * on robust accuracy, and outperforms it on clean accuracy. Note that when compared with APR_{SP}, it is even better on clean accuracy but slightly worse on robust accuracy.

5.4.5. Out-of-Distribution Detection Results

For OOD detection, we take CIFAR-10 as the in-distribution dataset and use a ResNet18 model trained on it to detect OOD samples. We compare our results with several configurations, such as training with cross-entropy, SupCLR [213] and CSI [208], as well as augmentation methods as Cutout, Mixup and APR. Note that we train with CE as well.

First of all, all our variants comfortably beat the baseline OOD detection (CE), which shows that our proposed method is indeed useful in increasing robustness. Furthermore, we see that our proposed methods are highly competitive, and they perform as good as the alternative methods. $HA_P^{++} + APR_S$ outperforms all other methods on LSUN and ImageNet datasets, and produces competitive results on others. Mean AUROC across

	OOD Datasets						
Method	SVHN	LSUN	ImageNet	LSUN (fix)	ImageNet (fix)	CIFAR-100	Mean
CE	88.6	90.7	88.3	87.5	87.4	85.8	88.1
CE + CutOut	93.6	94.5	90.2	92.2	89.0	86.4	91.0
CE + Mixup	78.1	80.7	76.5	80.7	76.0	74.9	77.8
SupCLR	97.3	92.8	91.4	91.6	90.5	88.6	92.0
CSI	96.5	96.3	96.2	92.1	92.4	**90.5**	94.0
CE + APR$_S$	90.4	96.1	94.2	90.9	89.1	86.8	91.3
CE + APR$_P$	**98.1**	93.7	95.2	91.4	91.1	88.9	93.1
CE + APR$_{PS}$	97.7	97.9	96.3	**93.7**	**92.8**	89.5	**94.7**
HA$_S$	93.0	96.3	93.6	91.5	90.4	87.4	92.0
HA$_P$	84.9	92.8	90.0	90.5	89.1	86.9	89.0
HA$_{PS}$	95.9	97.8	95.4	91.4	90.9	87.8	93.2
HA$_P^{++}$	92.7	92.2	91.0	89.6	89.4	86.2	90.2
HA$_S^{++}$	94.7	97.9	96.5	91.3	89.8	86.8	92.8
HA$_P^{++} + APR_S$	97.5	**98.7**	**97.8**	93.0	91.8	89.2	**94.7**
HA$_P^{++} + H_S$	96.9	98.3	97.1	90.6	89.9	86.4	93.2
HA$_{PS}^{++}$	96.6	98.7	97.7	93.0	91.2	88.1	94.2

Table 5.5 Out-of-distribution AUROC results on multiple datasets.

all datasets show that it ties with the best model APR_{PS}, showing its efficiency. The broader framework we propose leads to many variants with various performance profiles across different datasets, highlighting the flexibility and usefulness of our unification of frequency-centric augmentations. Note that the clean test accuracy values on CIFAR-10 (in-distribution dataset) are provided in Table 5.2, and shows that we perform the same or better than the other methods.

5.5. Conclusion

In this chapter, inspired by the frequency-centric explanations of how CNNs generalize, we propose two augmentations methods *HybridAugment* and *HybridAugment*++. The former aims to reduce the reliance of CNN generalization on high-frequency information in images, whereas the latter does the same but also promotes the use of phase information rather than the amplitude component. This unification of two distinct frequency-based analyses into a data augmentation method leads to competitive to or better than state-of-the-art methods on clean accuracy, corruption performance, adversarial performance and out-of-distribution detection. Both our methods are easy to implement, requires no extra data, extra models or model ensembles.

6. CONCLUSION

This chapter concludes the thesis with a summary and outlines future research directions.

6.1. Summary

This thesis is borne out of a years-long endeavour aimed at answering two questions outlined in chapter 3.1.:

- What are the robustness characteristics of models trained with extremely imbalanced data (i.e. zero-shot learning)?
- Can we leverage frequency-analyses of images to produce a method that improves the robustness of models against distribution shifts?

We present a detailed answer to the first question in chapter 3.. By analysing various discriminative ZSL models against adversaries, we first show that ZSL models are severely prone to such intrusions. These analyses show the important effect of training data and model maturity. Furthermore, the discrepancies between seen and unseen class behaviour is shown to be an integral part of the discussion, as we identify and define the *pseudo-robustness effect* that *masks* the robustness performance of *weak* models. We show that this effect leads to incorrect insights with regards to model robustness, and is of great importance when analysing any *weak* model, not just ZSL models, against adversaries.

Chapter 4. expands on *chapter* 3. and presents a common image corruption analyses of discriminative ZSL models. We first curate and present not one but three datasets aimed at analysing the robustness of any ZSL model. These datasets are the first of their kind, and are already released to the research community. Using these datasets, we perform our analyses with an even larger family of discriminative ZSL models and show that our findings apply to a broader segment of the literature. When coupled with key data augmentation methods, we obtain improved versions of existing discriminative ZSL models. We finally wrap up this chapter with a thorough discussion on and a comparison with the findings of chapter 3., which highlights the key differences of adversaries and common corruptions, and further highlights the need for a joint solution.

Chapter 5. focuses on developing methods to improve model robustness to distribution shifts. Inspired by the data-augmentation approaches, we formulate *HybridAugment* that creates frequency-swapped images on-the-fly and reduces the reliance of the models on high-frequency content. Doing so improves the robustness of the trained model, while keeping the clean accuracy values high. We then set our eyes on the bigger prize, which is to unify two orthogonal frequency-centric analyses (frequency bands and phase/amplitude) into one; we do so by proposing *HybridAugment*++, which performs hierarchical augmentations in frequency-spectra, first on frequency bands and then phase/amplitude components. *HybridAugment*++ comfortably outperforms other methods on multiple benchmarks on various distribution shifts, and also improves clean accuracy on multiple datasets.

6.2. Future Work

The advances reported in our thesis pave the way for new questions, and lead to several new exciting directions. For brevity and clarity, we list them below.

- The robust generalization perspective encompasses zero-shot learning as well, since unseen classes are a kind of out-of-distribution sample. Our thesis lays the groundwork and calls for their *grand* unification; we envision generic solutions that can be robust to distribution shifts while handling unseen classes successfully.
- Similar to above, we call for a broader approach to the robust generalization problem; instead of methods targeting specific distribution shifts, developing methods which improve robustness to *any* distribution shift should be the priority.
- There has been a trend on moving common image corruption simulations to more realistic/grounded frameworks, such as simulating corruptions taking into account the

scene geometry [135]. We think creating a dataset for ZSL with similar effects can take the discussions even further and benefit the community.

- Similar to above, targeting methods to alleviate such grounded corruptions, or even testing *HybridAugment* variants on them can be interesting.
- Specifically, *HybridAugment* variants can be combined with other augmentation methods and trained with better losses. This is likely to further improve its performance, and we believe this is a natural way forward.
- A natural extension of *HybridAugment* is to *learn* the cut-off frequency that divides high and low-frequency bands. Learning this from data can improve the results.
- Chapters 3. and 4., as detailed as they are, arguably only scratch the surface of the grand discussion of robust generalization in more practical scenarios, such as data imbalance, weak models and imperfect/incomplete supervision. We call for thorough endeavors towards this direction, as it might have significant impact on real-life adoption of ML systems.
- Finally, we call for a more thorough adoption of *the pseudo-robustness effect* in the literature for robustness evaluation against adversaries. Such adoption will provide better insights regarding model robustness, and might lead to better solutions towards achieving robust models.

’ There will be times when the struggle seems impossible. I know this already. Alone, unsure, dwarfed by the scale of the enemy. Remember this. Freedom is a pure idea. It occurs spontaneously and without instruction. ***One single thing will break the siege. Remember this. Try.****’*

Nemik’s Manifesto

Appendix A

Mean Corruption Errors

We provide mean corruption error (mCE) metrics for our corruption robustness experiments (see chapter 4.). These results are complementary to the relative, corruption-induced accuracy reduction ratios presented in the main manuscript and are given to establish a numerical baseline. We follow the calculation described in [9], but instead of AlexNet errors, we use the error of the original ALE model used in our thesis. Corruption error (CE) is calculated by

$$CE_c^f = \left(\sum_{s=1}^{5} E_{s,c}^f \right) \Bigg/ \left(\sum_{s=1}^{5} E_{s,c}^{ALE} \right) \quad (1)$$

where E is the error for our network f or ALE, s is the severity level and c is the corruption type. MCE is simply the average of CE values for 15 corruption types. Note that CE values for each corruption type and thus mCE for ALE will be 100. Therefore, any CE value or mCE value below 100 will mean the method being evaluated does better than ALE, or vice versa.

These average errors are given in Tables A.1, A.2, A.3 for AWA2-C, CUB-C and SUN-C datasets, respectively. We provide MCE metrics based on ZSL top-1 scores in Tables A.4, A.8, A.12, GZSL unseen scores in Tables A.5, A.9, A.13, GZSL seen scores in Tables A.6, A.10, A.14 and GZSL harmonic scores in Tables A.7, A.11, A.15 for AWA2-C, CUB-C and SUN-C datasets, respectively.

In Tables A.4 to A.15, *OR* refers to the original ALE model, *SS* refers to spatial smoothing, *TVM* refers to total variance minimization and *LS* refers to label smoothing defenses. We note that since spatial smoothing and total variance minimization are preprocessing defenses, their *clean* (uncorrupted) errors are the same as the original model. Corruption types given in the Tables are (from left to right) Gaussian noise, shot noise, impulse noise, defocus blur, glass blur, motion blur, zoom blur, snow, frost, fog, brightness, contrast, elastic transformations, pixelate and JPEG compression corruptions.

	Noise			**Blur**				**Weather**				**Digital**			
AWA2-C	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixelate	JPEG
ZSL	51.9	52.7	55.0	59.4	59.4	56.5	61.0	64.8	53.6	52.2	38.8	63.3	51.1	50.3	42.3
Unseen	93.3	94.1	94.8	92.9	94.0	91.8	96.3	95.4	92.8	92.3	85.8	95.0	91.7	90.7	89.2
Seen	46.9	50.3	53.1	65.1	61.7	55.6	71.6	65.5	52.0	44.8	24.5	63.3	39.7	42.1	29.8
H-Score	88.1	89.5	90.6	88.3	89.6	86.1	93.5	91.9	87.4	86.5	76.1	91.3	85.5	84.1	81.3

Table A.1 AWA2-C average errors per corruption category.

	Noise			**Blur**				**Weather**				**Digital**			
CUB-C	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixelate	JPEG
ZSL	77.0	77.6	80.8	63.2	71.6	61.0	65.5	77.5	71.1	65.0	53.3	72.9	59.8	61.0	59.3
Unseen	90.8	91.4	93.0	86.8	89.3	84.9	86.2	90.8	88.9	87.7	77.9	90.6	82.4	83.5	83.3
Seen	76.9	78.8	82.0	64.5	72.6	60.6	67.0	81.2	73.4	61.7	47.3	70.6	57.2	62.3	57.1
H-Score	86.8	87.8	89.9	80.8	84.7	78.2	80.6	87.7	84.3	81.4	68.9	85.8	75.0	77.1	76.0

Table A.2 CUB-C average errors per corruption category.

	Noise			**Blur**				**Weather**				**Digital**			
SUN-C	Gaussian	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixelate	JPEG
ZSL	76.0	79.1	81.0	63.1	79.0	67.2	63.5	79.5	75.5	57.2	50.7	65.1	70.2	61.2	54.9
Unseen	93.3	94.1	94.8	89.6	93.8	90.8	93.3	95.0	94.0	85.7	83.2	89.4	91.1	87.8	85.3
Seen	90.4	91.7	93.4	84.5	90.8	86.3	90.1	93.5	90.0	79.5	75.7	84.0	87.2	83.6	78.3
H-Score	92.2	93.1	94.2	87.6	92.7	89.0	92.0	94.4	92.5	83.1	80.1	87.3	89.5	86.0	82.5

Table A.3 SUN-C average errors per corruption category.

			Noise			**Blur**				**Weather**				**Digital**			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	38.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	38.0	106.6	121.0	118.7	90.9	104.2	102.7	105.0	106.8	108.3	108.9	109.2	106.2	104.7	102.8	103.6	106.5
TVM	38.0	102.7	99.8	100.0	103.3	100.5	100.1	101.7	101.9	106.3	103.3	106.3	105.2	104.3	103.1	101.0	103.4
LS	39.4	104.6	103.1	104.6	104.7	102.6	103.0	107.1	103.0	103.6	106.2	108.1	104.7	105.4	102.8	105.2	104.9
AM	55.1	107.0	109.6	108.6	106.4	101.8	103.0	104.9	106.9	101.8	104.0	104.8	119.0	102.5	107.4	107.5	116.5
ANT	54.9	104.9	100.5	101.0	98.1	101.8	102.4	105.8	106.7	102.3	108.0	105.4	118.7	104.2	104.9	103.3	110.7

Table A.4 AWA2-C mCE values based on ZSL top-1 accuracy.

			Noise			**Blur**				**Weather**				**Digital**			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	84.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	84.7	100.9	101.8	102.0	96.2	100.6	100.6	101.2	100.4	102.1	101.4	101.7	101.7	101.0	101.3	100.5	100.5
TVM	84.7	100.9	100.5	100.5	100.6	100.2	100.4	100.7	99.8	101.3	101.2	101.8	102.6	101.0	101.1	100.2	100.9
LS	83.7	98.2	98.5	98.7	98.1	98.1	98.9	98.6	98.3	98.1	98.5	97.6	97.6	98.6	98.6	97.2	97.9
AM	83.9	98.5	96.6	96.5	96.9	99.6	97.1	99.0	97.3	100.9	100.4	99.2	98.6	98.1	99.3	99.4	98.3
ANT	80.7	95.3	90.5	89.5	91.3	98.7	96.5	98.2	97.8	98.9	95.9	95.7	94.7	95.9	95.3	96.1	94.3

Table A.5 AWA2-C mCE values based on unseen accuracy.

			Noise			**Blur**				**Weather**				**Digital**			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	21.2	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	21.2	113.7	143.0	133.0	74.0	106.1	105.4	112.8	107.9	117.1	117.2	120.8	122.2	110.1	112.6	107.6	115.6
TVM	21.2	106.7	113.3	109.9	107.5	100.7	100.1	103.6	100.7	111.6	105.4	114.9	111.9	106.5	107.5	101.5	104.9
LS	25.8	106.1	107.4	105.8	105.6	103.1	103.2	104.6	101.6	102.3	104.7	105.9	117.1	103.2	107.3	107.0	113.1
AM	16.5	86.2	87.5	86.1	86.8	90.3	94.4	87.1	90.6	89.4	84.2	78.3	78.4	89.3	83.6	86.2	81.1
ANT	14.6	83.8	71.3	70.2	72.5	89.2	92.9	88.2	94.1	96.8	92.0	85.3	72.4	95.5	83.2	81.3	72.7

Table A.6 AWA2-C mCE values based on seen accuracy.

			Noise			**Blur**				**Weather**				**Digital**			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	74.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	74.3	101.7	103.6	103.7	93.4	101.2	101.1	102.3	100.8	103.7	102.5	103.1	103.0	101.8	102.3	100.9	101.1
TVM	74.3	101.5	101.1	100.9	101.2	100.3	100.7	101.3	99.8	102.4	102.1	103.3	104.4	101.8	102.0	100.4	101.6
LS	73.2	97.4	97.7	98.0	97.0	97.7	98.4	98.2	97.5	97.3	97.8	96.4	96.6	98.1	97.9	95.8	96.9
AM	73.0	97.4	94.4	94.2	94.8	98.9	95.5	98.1	95.6	101.5	100.5	98.4	97.5	96.9	98.7	98.9	97.0
ANT	68.4	92.4	84.9	83.5	86.2	97.4	94.6	96.9	96.4	98.3	93.6	93.0	91.2	94.1	92.2	93.5	90.5

Table A.7 AWA2-C mCE values based on harmonic accuracy.

			Noise			**Blur**				**Weather**				**Digital**			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	45.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	45.5	106.3	107.1	107.2	86.4	106.2	104.6	105.8	106.5	110.7	113.5	114.2	107.2	106.6	106.5	107.2	104.5
TVM	45.5	100.8	96.7	98.0	99.5	101.1	99.4	100.5	101.2	102.6	98.9	104.5	104.7	102.6	103.0	100.5	99.0
LS	47.8	101.3	104.1	105.2	102.8	103.0	99.6	98.5	98.4	95.4	99.7	106.2	98.2	101.6	99.8	102.5	104.7
AM	48.4	102.1	101.7	102.8	101.5	101.1	104.4	101.6	99.0	103.5	101.4	99.1	104.2	100.3	102.9	103.0	105.2
ANT	51.1	100.8	97.3	99.9	97.8	105.0	100.9	104.7	102.3	94.1	95.8	99.7	106.5	99.4	99.8	102.3	105.7

Table A.8 CUB-C mCE values based on ZSL top-1 accuracy.

			Noise			**Blur**				**Weather**				**Digital**			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	74.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	74.4	103.0	104.3	103.9	95.3	102.5	101.9	103.1	102.8	104.2	104.5	104.2	105.4	102.2	103.3	103.5	103.2
TVM	74.4	100.3	99.5	99.4	99.7	100.5	99.1	100.6	100.7	100.7	100.1	101.5	101.6	100.5	101.5	100.2	99.3
LS	77.3	101.8	102.6	102.2	102.0	101.8	99.7	101.9	102.1	99.3	101.3	102.8	101.5	102.7	101.4	102.4	103.1
AM	72.8	98.3	99.5	99.4	99.1	97.8	98.8	97.8	97.5	99.8	98.2	97.6	96.4	98.5	98.7	97.7	98.7
ANT	73.9	97.2	96.2	96.9	96.1	99.0	99.2	97.2	97.2	96.7	95.7	96.6	99.1	98.0	97.2	96.8	96.8

Table A.9 CUB-C mCE values based on unseen accuracy.

			Noise			Blur				Weather				Digital			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	35.4	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	35.4	109.2	111.2	108.9	84.2	108.2	107.5	110.4	110.1	110.1	113.8	120.4	113.5	110.3	111.5	110.8	107.9
TVM	35.4	102.1	99.3	99.3	99.8	102.1	99.9	101.8	100.3	103.2	99.7	105.6	109.2	104.0	105.7	101.7	99.2
LS	43.8	105.7	105.6	106.0	104.4	108.4	101.1	105.7	104.8	96.3	102.9	110.9	112.6	107.3	104.7	104.5	109.7
AM	39.9	103.8	103.5	103.3	102.3	102.0	103.8	103.0	101.8	104.3	104.4	99.3	108.2	100.8	106.6	105.0	109.2
ANT	39.4	101.8	97.7	98.9	97.2	103.0	105.6	102.9	104.0	94.7	94.1	99.7	109.9	100.9	102.1	102.4	113.5

Table A.10 CUB-C mCE values based on seen accuracy.

			Noise			Blur				Weather				Digital			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	63.3	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	63.3	104.5	106.3	105.5	92.7	103.9	103.2	104.8	104.5	105.9	106.8	107.0	107.8	103.6	105.1	105.3	104.6
TVM	63.3	100.6	99.4	99.2	99.6	101.0	99.1	100.9	100.8	101.3	100.0	102.4	103.0	101.0	102.4	100.5	99.1
LS	67.6	102.7	103.7	103.5	102.8	103.0	99.8	102.8	103.0	98.6	101.8	104.5	103.3	104.0	102.2	103.3	104.6
AM	62.5	98.8	100.0	100.0	99.5	97.9	99.4	98.1	97.8	100.8	99.1	96.0	97.3	98.3	99.6	98.7	99.9
ANT	63.6	97.4	96.0	97.1	96.0	99.3	100.1	97.4	97.9	95.8	94.7	96.0	100.4	97.8	97.4	97.3	97.6

Table A.11 CUB-C mCE values based on harmonic accuracy.

			Noise			Blur				Weather				Digital			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	42.6	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	42.6	104.9	107.6	105.8	83.4	106.1	102.0	103.8	104.9	109.4	108.0	116.8	104.7	107.6	102.5	104.4	106.0
TVM	42.6	99.8	97.0	95.9	98.5	100.2	98.0	98.0	97.5	101.6	100.0	105.5	103.8	102.9	99.9	99.9	98.4
LS	44.8	101.2	100.6	99.6	100.5	102.0	96.9	102.0	99.8	98.8	99.0	106.5	101.3	105.1	99.7	103.4	102.7
AM	41.6	100.4	101.4	99.9	101.8	101.6	98.0	99.5	100.8	100.0	99.2	102.5	98.0	99.0	98.0	104.5	102.3
ANT	42.6	96.5	92.7	93.0	92.3	101.9	99.8	100.2	96.3	97.5	98.4	96.3	94.0	95.6	96.7	95.6	97.5

Table A.12 SUN-C mCE values based on Top-1 ZSL accuracy.

			Noise			Blur				Weather				Digital			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	79.5	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	79.5	100.7	102.3	102.0	94.3	100.3	100.1	100.4	100.5	102.0	101.2	103.9	101.0	101.4	99.9	100.6	100.4
TVM	79.5	100.1	99.6	100.0	100.5	100.0	99.0	99.5	99.2	100.4	99.6	101.8	101.2	100.8	100.3	100.3	99.6
LS	81.6	101.4	100.9	101.0	101.6	103.3	101.7	101.3	100.4	99.5	100.4	102.5	102.5	101.8	101.1	101.8	101.8
AM	76.4	98.8	99.2	99.6	99.2	99.6	99.8	98.1	97.9	99.3	99.4	98.9	96.4	100.0	99.6	98.8	96.8
ANT	76.4	98.5	97.0	97.1	97.6	99.6	99.9	99.3	100.2	99.6	100.5	98.1	97.0	98.5	99.0	97.6	96.8

Table A.13 SUN-C mCE values based on unseen accuracy.

			Noise			Blur				Weather				Digital			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	67.7	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	67.7	102.5	104.6	103.4	93.0	102.8	101.1	101.2	101.9	102.9	103.4	108.0	103.3	104.2	102.1	102.3	103.1
TVM	67.7	100.7	100.3	100.1	100.4	100.0	100.0	99.3	99.6	100.7	100.8	103.6	102.4	102.0	100.9	100.3	100.7
LS	68.4	100.0	100.1	99.9	99.9	102.4	100.4	100.0	100.1	98.7	99.1	101.1	98.5	100.4	100.3	98.8	99.9
AM	64.3	98.7	99.6	99.5	1002	98.4	99.2	98.3	99.4	99.4	99.8	99.5	94.2	99.5	98.5	98.3	96.9
ANT	64.2	97.1	96.2	96.2	96.5	98.6	99.7	98.1	98.4	98.8	99.3	97.1	92.4	96.7	98.1	94.7	95.5

Table A.14 SUN-C mCE values based on seen accuracy.

			Noise			**Blur**				**Weather**				**Digital**			
Model	Clean	**mCE**	Gauss	Shot	Imp.	Defo.	Glass	Mot.	Zoom	Snow	Frost	Fog	Brigh.	Cont.	Elas.	Pixel	JPEG
OR	74.9	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0	100.0
SS	74.9	101.3	103.2	102.6	93.7	101.2	100.4	100.7	101.0	102.4	102.0	105.4	101.8	102.3	100.7	101.2	101.3
TVM	74.9	100.3	99.8	100.0	100.5	100.0	99.4	00.4	99.3	100.5	100.0	102.5	101.6	101.2	100.5	100.3	99.9
LS	76.7	101.1	100.7	100.7	101.1	103.3	101.5	101.0	100.4	99.2	100.1	102.2	101.5	101.6	101.0	100.9	101.4
AM	71.6	98.8	99.4	99.7	99.7	99.2	99.6	98.1	98.4	99.3	99.5	99.1	95.6	99.8	99.3	98.6	96.7
ANT	71.6	97.9	96.6	96.7	97.1	99.3	99.8	98.9	98.3	99.3	100.3	97.7	95.4	97.9	98.7	96.5	96.3

Table A.15 SUN-C mCE values based on harmonic accuracy.

REFERENCES

[1] Yang Liu, Jishun Guo, Deng Cai, and Xiaofei He. Attribute attention for semantic disambiguation in zero-shot learning. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6698–6707. **2019**.

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

[3] Zewen Li, Fan Liu, Wenjie Yang, Shouheng Peng, and Jun Zhou. A survey of convolutional neural networks: analysis, applications, and prospects. *IEEE transactions on neural networks and learning systems*, **2021**.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. **2016**.

[5] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500. **2017**.

[6] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708. **2017**.

[7] Farhad Pourpanah, Moloud Abdar, Yuxuan Luo, Xinlei Zhou, Ran Wang, Chee Peng Lim, Xi-Zhao Wang, and QM Jonathan Wu. A review of generalized zero-shot learning methods. *IEEE transactions on pattern analysis and machine intelligence*, **2022**.

[8] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *arXiv preprint arXiv:1811.12231*, **2018**.

[9] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, **2019**.

[10] Guangyao Chen, Peixi Peng, Li Ma, Jia Li, Lin Du, and Yonghong Tian. Amplitude-phase recombination: Rethinking robustness of convolutional neural networks in frequency domain.

In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 458–467. **2021**.

[11] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626. **2017**.

[12] Filip Štetić. The history of computer vision and the evolution of autonomous vehicles, **2022**.

[13] Lawrence G Roberts. *Machine perception of three-dimensional solids*. Ph.D. thesis, Massachusetts Institute of Technology, **1963**.

[14] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, **1988**.

[15] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001*, volume 1, pages I–I. Ieee, **2001**.

[16] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, **2004**.

[17] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, **1998**.

[18] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, **1995**.

[19] Daniel Scharstein and Chris Pal. Learning conditional random fields for stereo. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, **2007**.

[20] Juergen Gall, Nima Razavi, and Luc Van Gool. An introduction to random forests for multi-class object detection. In *Outdoor and large-scale real-world scene analysis*, pages 243–263. Springer, **2012**.

[21] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, **2006**.

[22] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *2011 International conference on computer vision*, pages 2564–2571. Ieee, **2011**.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, **2017**.

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

[25] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, **2017**.

[26] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, **2020**.

[27] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), **2015**.

[28] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717. **2020**.

[29] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, **2020**.

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

[31] Nicholas Carlini and David Wagner. Audio adversarial examples: Targeted attacks on speech-to-text. In *2018 IEEE security and privacy workshops (SPW)*, pages 1–7. IEEE, **2018**.

[32] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*, **2017**.

[33] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349. **2021**.

[34] David Stutz, Matthias Hein, and Bernt Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987. **2019**.

[35] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, **2019**.

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

[37] Tonmoy Saikia, Cordelia Schmid, and Thomas Brox. Improving robustness against common corruptions with frequency biased models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10211–10220. **2021**.

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

[39] Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8684–8694. **2020**.

[40] Aude Oliva, Antonio Torralba, and Philippe G Schyns. Hybrid images. *ACM Transactions on Graphics (TOG)*, 25(3):527–532, **2006**.

[41] Frank Rosenblatt. The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, 65(6):386, **1958**.

[42] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, **1951**.

[43] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448. **2015**.

[44] Wufeng Xue, Lei Zhang, Xuanqin Mou, and Alan C Bovik. Gradient magnitude similarity deviation: A highly efficient perceptual image quality index. *IEEE transactions on image processing*, 23(2):684–695, **2013**.

[45] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, **1986**.

[46] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, **2014**.

[47] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, **2010**.

[48] Aristidis Likas, Nikos Vlassis, and Jakob J Verbeek. The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461, **2003**.

[49] Hakan Bilen and Andrea Vedaldi. Weakly supervised deep detection networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2846–2854. **2016**.

[50] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):189–203, **2016**.

[51] Konstantinos Georgiadis, Albert Saà-Garriga, Mehmet Kerim Yucel, Anastasios Drosou, and Bruno Manganelli. Adaptive mask-based pyramid network for realistic bokeh rendering. *arXiv preprint arXiv:2210.16078*, **2022**.

[52] Ravi Garg, Vijay Kumar Bg, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European conference on computer vision*, pages 740–756. Springer, **2016**.

[53] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *European conference on computer vision*, pages 69–84. Springer, **2016**.

[54] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)*, 53(3):1–34, **2020**.

[55] Wei Wang, Vincent W Zheng, Han Yu, and Chunyan Miao. A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–37, **2019**.

[56] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, **2014**.

[57] Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. Highway networks. *arXiv preprint arXiv:1505.00387*, **2015**.

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

[59] Yongqin Xian, Bernt Schiele, and Zeynep Akata. Zero-shot learning-the good, the bad and the ugly. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4582–4591. **2017**.

[60] Fangxin Wang, Jie Liu, Shuwu Zhang, Guixuan Zhang, Yuejun Li, and Fei Yuan. Inductive zero-shot image annotation via embedding graph. *IEEE Access*, 7:107816–107830, **2019**.

[61] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. **2010**.

[62] Genevieve Patterson, Chen Xu, Hang Su, and James Hays. The sun attribute database: Beyond categories for deeper scene understanding. *International Journal of Computer Vision*, 108(1):59–81, **2014**.

[63] Wei-Lun Chao, Soravit Changpinyo, Boqing Gong, and Fei Sha. An empirical study and analysis of generalized zero-shot learning for object recognition in the wild. In *European conference on computer vision*, pages 52–68. Springer, **2016**.

[64] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *2009 IEEE conference on computer vision and pattern recognition*, pages 951–958. IEEE, **2009**.

[65] Christoph H Lampert, Hannes Nickisch, and Stefan Harmeling. Attribute-based classification for zero-shot visual object categorization. *IEEE transactions on pattern analysis and machine intelligence*, 36(3):453–465, **2013**.

[66] Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. Zero-shot learning through cross-modal transfer. *Advances in neural information processing systems*, 26, **2013**.

[67] Bo Zhao, Xinwei Sun, Yuan Yao, and Yizhou Wang. Zero-shot learning via shared-reconstruction-graph pursuit. *arXiv preprint arXiv:1711.07302*, **2017**.

[68] Yinduo Wang, Haofeng Zhang, Zheng Zhang, and Yang Long. Asymmetric graph based zero shot learning. *Multimedia Tools and Applications*, 79(45):33689–33710, **2020**.

[69] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208. **2018**.

[70] Vinay Kumar Verma, Ashish Mishra, Anubha Pandey, Hema A Murthy, and Piyush Rai. Towards zero-shot learning with fewer seen class examples. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2241–2251. **2021**.

[71] Dat Huynh and Ehsan Elhamifar. Fine-grained generalized zero-shot learning via dense attribute-based attention. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4483–4493. **2020**.

[72] Senthil Purushwalkam, Maximilian Nickel, Abhinav Gupta, and Marc'Aurelio Ranzato. Task-driven modular networks for zero-shot compositional learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3593–3602. **2019**.

[73] Yashas Annadani and Soma Biswas. Preserving semantic relations for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7603–7612. **2018**.

[74] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Transferable contrastive network for generalized zero-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9765–9774. **2019**.

[75] Huajie Jiang, Ruiping Wang, Shiguang Shan, and Xilin Chen. Learning class prototypes via structure alignment for zero-shot recognition. In *Proceedings of the European conference on computer vision (ECCV)*, pages 118–134. **2018**.

[76] Rafael Felix, Ian Reid, Gustavo Carneiro, et al. Multi-modal cycle-consistent generalized zero-shot learning. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 21–37. **2018**.

[77] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289. **2018**.

[78] Maxime Bucher, Stéphane Herbin, and Frédéric Jurie. Generating visual representations for zero-shot classification. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 2666–2673. **2017**.

[79] Mert Bulent Sariyildiz and Ramazan Gokberk Cinbis. Gradient matching generative networks for zero-shot learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2168–2178. **2019**.

[80] Meng Ye and Yuhong Guo. Zero-shot classification with discriminative semantic representation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7140–7148. **2017**.

[81] Xing Xu, Fumin Shen, Yang Yang, Jie Shao, and Zi Huang. Transductive visual-semantic embedding for zero-shot learning. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pages 41–49. **2017**.

[82] Yan Li, Junge Zhang, Jianguo Zhang, and Kaiqi Huang. Discriminative learning of latent features for zero-shot recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7463–7471. **2018**.

[83] Zeynep Akata, Florent Perronnin, Zaid Harchaoui, and Cordelia Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 819–826. **2013**.

[84] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. Zero-shot recognition via semantic embeddings and knowledge graphs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6857–6866. **2018**.

[85] Li Zhang, Tao Xiang, and Shaogang Gong. Learning a deep embedding model for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2021–2030. **2017**.

[86] Ziming Zhang and Venkatesh Saligrama. Zero-shot learning via semantic similarity embedding. In *Proceedings of the IEEE international conference on computer vision*, pages 4166–4174. **2015**.

[87] Soravit Changpinyo, Wei-Lun Chao, Boqing Gong, and Fei Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5327–5336. **2016**.

[88] Yongqin Xian, Tobias Lorenz, Bernt Schiele, and Zeynep Akata. Feature generating networks for zero-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5542–5551. **2018**.

[89] Guo-Sen Xie, Zheng Zhang, Guoshuai Liu, Fan Zhu, Li Liu, Ling Shao, and Xuelong Li. Generalized zero-shot learning with multiple graph adaptive generative networks. *IEEE transactions on neural networks and learning systems*, **2021**.

[90] Yuanbo Ma, Xing Xu, Fumin Shen, and Heng Tao Shen. Similarity preserving feature generating networks for zero-shot learning. *Neurocomputing*, 406:333–342, **2020**.

[91] Jingjing Li, Mengmeng Jing, Ke Lu, Zhengming Ding, Lei Zhu, and Zi Huang. Leveraging the invariant side of generative zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7402–7411. **2019**.

[92] Rui Gao, Xingsong Hou, Jie Qin, Jiaxin Chen, Li Liu, Fan Zhu, Zhao Zhang, and Ling Shao. Zero-vae-gan: Generating unseen features for generalized and transductive zero-shot learning. *IEEE Transactions on Image Processing*, 29:3665–3680, **2020**.

[93] Yongqin Xian, Saurabh Sharma, Bernt Schiele, and Zeynep Akata. f-vaegan-d2: A feature generating framework for any-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10275–10284. **2019**.

[94] Chuanlong Li, Xiufen Ye, Haibo Yang, Yatong Han, Xiang Li, and Yunpeng Jia. Generalized zero shot learning via synthesis pseudo features. *IEEE Access*, 7:87827–87836, **2019**.

[95] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271. **2021**.

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

[97] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. *Advances in neural information processing systems*, 29, **2016**.

[98] Ekin D Cubuk, Barret Zoph, Samuel S Schoenholz, and Quoc V Le. Intriguing properties of adversarial examples. *arXiv preprint arXiv:1711.02846*, **2017**.

[99] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, **2017**.

[100] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, **2018**.

[101] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, **2019**.

[102] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A Forsyth. Unrestricted adversarial examples via semantic manipulation. *arXiv preprint arXiv:1904.06347*, **2019**.

[103] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582. **2016**.

[104] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, **2016**.

[105] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773. **2017**.

[106] Tao Bai, Jun Zhao, Jinlin Zhu, Shoudong Han, Jiefeng Chen, Bo Li, and Alex Kot. Ai-gan: Attack-inspired generation of adversarial examples. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2543–2547. IEEE, **2021**.

[107] BS Vivek, Konda Reddy Mopuri, and R Venkatesh Babu. Gray-box adversarial training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 203–218. **2018**.

[108] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. *Ieee Access*, 6:14410–14430, **2018**.

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

[110] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1369–1378. **2017**.

[111] Ziqi Zhang, Xinge Zhu, Yingwei Li, Xiangqun Chen, and Yao Guo. Adversarial attacks on monocular depth estimation. *arXiv preprint arXiv:2003.10315*, **2020**.

[112] Siyuan Liang, Xingxing Wei, Siyuan Yao, and Xiaochun Cao. Efficient adversarial attacks for visual object tracking. In *European Conference on Computer Vision*, pages 34–50. Springer, **2020**.

[113] Daniel Liu, Ronald Yu, and Hao Su. Extending adversarial attacks and defenses to deep 3d point cloud classifiers. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2279–2283. IEEE, **2019**.

[114] Vasu Sharma, Ankita Kalra, Sumedha Chaudhary Vaibhav, Labhesh Patel, and Louis-Phillippe Morency. Attend and attack: Attention guided adversarial attacks on visual question answering models. In *Proc. 32nd Conf. Neural Inf. Process. Syst.(NeurIPS)*, pages 1–6. **2018**.

[115] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 41(8):1979–1993, **2018**.

[116] Tamir Hazan, George Papandreou, and Daniel Tarlow. *Perturbations, optimization, and statistics*. MIT Press, **2016**.

[117] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *Proceedings of the ieee conference on computer vision and pattern recognition*, pages 4480–4488. **2016**.

[118] Andrew Ross and Finale Doshi-Velez. Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. **2018**.

[119] Yoshua Bengio et al. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, **2009**.

[120] Jiajun Lu, Theerasit Issaranon, and David Forsyth. Safetynet: Detecting and rejecting adversarial examples robustly. In *Proceedings of the IEEE international conference on computer vision*, pages 446–454. **2017**.

[121] Tejas Borkar, Felix Heide, and Lina Karam. Defending against universal attacks through selective feature regeneration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 709–719. **2020**.

[122] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, **2016**.

[123] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, **2017**.

[124] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, **2017**.

[125] Uri Shaham, James Garritano, Yutaro Yamada, Ethan Weinberger, Alex Cloninger, Xiuyuan Cheng, Kelly Stanton, and Yuval Kluger. Defending against adversarial images using basis functions transformations. *arXiv preprint arXiv:1803.10840*, **2018**.

[126] Hyeungill Lee, Sungyeob Han, and Jungwoo Lee. Generative adversarial trainer: Defense to adversarial perturbations with gan. *arXiv preprint arXiv:1705.03387*, **2017**.

[127] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC conference on computer and communications security*, pages 135–147. **2017**.

[128] Jianli Zhou, Chao Liang, and Jun Chen. Manifold projection for adversarial defense on face recognition. In *European Conference on Computer Vision*, pages 288–305. Springer, **2020**.

[129] Jiang Liu, Alexander Levine, Chun Pong Lau, Rama Chellappa, and Soheil Feizi. Segment and complete: Defending object detectors against adversarial patch attacks with robust patch detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14973–14982. **2022**.

[130] Piotr Żelasko, Sonal Joshi, Yiwen Shao, Jesus Villalba, Jan Trmal, Najim Dehak, and Sanjeev Khudanpur. Adversarial attacks and defenses for speech recognition systems. *arXiv preprint arXiv:2103.17122*, **2021**.

[131] Xiaosen Wang, Jin Hao, Yichen Yang, and Kun He. Natural language adversarial defense through synonym encoding. In *Uncertainty in Artificial Intelligence*, pages 823–833. PMLR, **2021**.

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

[133] Jiachen Sun, Qingzhao Zhang, Bhavya Kailkhura, Zhiding Yu, and Z Morley Mao. Modelnet40-c: Arobustness benchmark for 3d point cloud recognition under corruption.

[134] Alfred Laugros, Alice Caplier, and Matthieu Ospici. Using synthetic corruptions to measure robustness to natural distribution shifts. *arXiv preprint arXiv:2107.12052*, **2021**.

[135] Oğuzhan Fatih Kar, Teresa Yeo, Andrei Atanov, and Amir Zamir. 3d common corruptions and data augmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18963–18974. **2022**.

[136] Christoph Kamann and Carsten Rother. Benchmarking the robustness of semantic segmentation models with respect to common corruptions. *International Journal of Computer Vision*, 129(2):462–483, **2021**.

[137] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, **2019**.

[138] Haotao Wang, Chaowei Xiao, Jean Kossaifi, Zhiding Yu, Anima Anandkumar, and Zhangyang Wang. Augmax: Adversarial composition of random augmentations for robust training. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*. **2021**.

[139] Jin-Ha Lee, Muhammad Zaigham Zaheer, Marcella Astrid, and Seung-Ik Lee. Smoothmix: A simple yet effective data augmentation to train robust classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. **2020**.

[140] Apostolos Modas, Rahul Rade, Guillermo Ortiz-Jiménez, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Prime: A few primitives can boost robustness to common corruptions. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 623–640. Springer Nature Switzerland, Cham, **2022**. ISBN 978-3-031-19806-9.

[141] Dan Hendrycks, Andy Zou, Mantas Mazeika, Leonard Tang, Bo Li, Dawn Song, and Jacob Steinhardt. Pixmix: Dreamlike pictures comprehensively improve safety measures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16783–16792. **2022**.

[142] Dan Andrei Calian, Florian Stimberg, Olivia Wiles, Sylvestre-Alvise Rebuffi, András György, Timothy A Mann, and Sven Gowal. Defending against image corruptions through adversarial augmentations. In *International Conference on Learning Representations*. **2022**.

[143] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., **2019**.

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

[145] Evgenia Rusak, Lukas Schott, Roland S. Zimmermann, Julian Bitterwolf, Oliver Bringmann, Matthias Bethge, and Wieland Brendel. A simple way to make neural networks robust against diverse image corruptions. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 53–69. Springer International Publishing, Cham, **2020**. ISBN 978-3-030-58580-8.

[146] Shruthi Gowda, Bahram Zonooz, and Elahe Arani. Inbiased: Inductive bias distillation to improve generalization and robustness through shape-awareness, **2022**. doi:10.48550/ARXIV.2206.05846.

[147] Mingjie Sun, Zichao Li, Chaowei Xiao, Haonan Qiu, Bhavya Kailkhura, Mingyan Liu, and Bo Li. Can shape structure features improve model robustness under diverse adversarial settings? In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7526–7535. **2021**.

[148] Hubert Lin, Mitchell van Zuijlen, Sylvia C. Pont, Maarten W.A. Wijntjes, and Kavita Bala. What can style transfer and paintings do for model robustness? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11028–11037. **2021**.

[149] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Revisiting batch normalization for improving corruption robustness. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 494–503. **2021**.

[150] Oren Nuriel, Sagie Benaim, and Lior Wolf. Permuted adain: Reducing the bias towards global statistics in image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9482–9491. **2021**.

[151] Teresa Yeo, Oğuzhan Fatih Kar, and Amir Zamir. Robustness via cross-domain ensembles. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12189–12199. **2021**.

[152] Tonmoy Saikia, Cordelia Schmid, and Thomas Brox. Improving robustness against common corruptions with frequency biased models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10211–10220. **2021**.

[153] Yong Guo, David Stutz, and Bernt Schiele. Improving robustness by enhancing weak subnets, **2022**. doi:10.48550/ARXIV.2201.12765.

[154] Kyle Otstot, John Kevin Cava, Tyler Sypherd, and Lalitha Sankar. Augloss: A learning methodology for real-world dataset corruption, **2022**. doi:10.48550/ARXIV.2206.02286.

[155] Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. Wavelet integrated cnns for noise-robust image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. **2020**.

[156] Alvin Chan, Yew-Soon Ong, and Clement Tan. How does frequency bias affect the robustness of neural image classifiers against common corruption and adversarial perturbations?, **2022**. doi:10.48550/ARXIV.2205.04533.

[157] Jiachen Sun, Akshay Mehra, Bhavya Kailkhura, Pin-Yu Chen, Dan Hendrycks, Jihun Hamm, and Z. Morley Mao. A spectral view of randomized smoothing under common corruptions: Benchmarking and improving certified robustness. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 654–671. Springer Nature Switzerland, Cham, **2022**. ISBN 978-3-031-19772-7.

[158] Yuyang Long, Qilong Zhang, Boheng Zeng, Lianli Gao, Xianglong Liu, Jian Zhang, and Jingkuan Song. Frequency domain model augmentation for adversarial attack. In Shai Avidan, Gabriel Brostow, Moustapha Cissé, Giovanni Maria Farinella, and Tal Hassner, editors, *Computer Vision – ECCV 2022*, pages 549–566. Springer Nature Switzerland, Cham, **2022**. ISBN 978-3-031-19772-7.

[159] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 7472–7482. PMLR, **2019**.

[160] Ronald Newbold Bracewell and Ronald N Bracewell. *The Fourier transform and its applications*, volume 31999. McGraw-Hill New York, **1986**.

[161] William B. Pennebaker and Joan L. Mitchell. *JPEG Still Image Data Compression Standard*. Van Nostrand Reinhold, New York, **1992**.

[162] Lu Chi, Borui Jiang, and Yadong Mu. Fast fourier convolution. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 4479–4488. Curran Associates, Inc., **2020**.

[163] Yulong Cao, Chaowei Xiao, Dawei Yang, Jing Fang, Ruigang Yang, Mingyan Liu, and Bo Li. Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*, **2019**.

[164] Xiaoyong Yuan, Pan He, Qile Zhu, and Xiaolin Li. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9):2805–2824, **2019**.

[165] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. **2017**.

[166] Jason Weston, Samy Bengio, and Nicolas Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35, **2010**.

[167] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, **2016**.

[168] Vahid Behzadan and Arslan Munir. Vulnerability of deep reinforcement learning to policy induction attacks. In *International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 262–275. Springer, **2017**.

[169] Daniel Zügner and Stephan Günnemann. Adversarial attacks on graph neural networks via meta learning. *arXiv preprint arXiv:1902.08412*, **2019**.

[170] Anurag Arnab, Ondrej Miksik, and Philip HS Torr. On the robustness of semantic segmentation models to adversarial attacks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 888–897. **2018**.

[171] Xuesong Chen, Xiyu Yan, Feng Zheng, Yong Jiang, Shu-Tao Xia, Yong Zhao, and Rongrong Ji. One-shot adversarial attacks on visual tracking with dual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10176–10185. **2020**.

[172] Yunhan Jia, Yantao Lu, Junjie Shen, Qi Alfred Chen, Hao Chen, Zhenyu Zhong, and Tao Wei. Fooling detection alone is not enough: Adversarial attack against multiple object tracking. In *International Conference on Learning Representations*. **2019**.

[173] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *CoRR*, abs/1611.02770, **2016**.

[174] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 262–271. **2020**.

[175] Chengzhi Mao, Ziyuan Zhong, Junfeng Yang, Carl Vondrick, and Baishakhi Ray. Metric learning for adversarial robustness. In *Advances in Neural Information Processing Systems*, pages 478–489. **2019**.

[176] Han Xu, Yao Ma, Haochen Liu, Debayan Deb, H. S. Liu, Jiliang Tang, and Anil Jain. Adversarial attacks and defenses in images, graphs and text: A review. *ArXiv*, abs/1909.08072, **2019**.

[177] L. Jing and Y. Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, **2020**.

[178] Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Marc' Aurelio Ranzato, and Tomas Mikolov. Devise: A deep visual-semantic embedding model. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2121–2129. Curran Associates, Inc., **2013**.

[179] Zeynep Akata, Scott Reed, Daniel Walter, Honglak Lee, and Bernt Schiele. Evaluation of output embeddings for fine-grained image classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2015**.

[180] Yongqin Xian, Zeynep Akata, Gaurav Sharma, Quynh Nguyen, Matthias Hein, and Bernt Schiele. Latent embeddings for zero-shot classification. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2016**.

[181] Long Chen, Hanwang Zhang, Jun Xiao, Wei Liu, and Shih-Fu Chang. Zero-shot visual recognition using semantics-preserving adversarial embedding networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. **2018**.

[182] Yunlong Yu, Zhong Ji, Yanwei Fu, Jichang Guo, Yanwei Pang, and Zhongfei (Mark) Zhang. Stacked semantics-guided attention model for fine-grained zero-shot learning. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 5995–6004. **2018**.

[183] Haofeng Zhang, Yang Long, Yu Guan, and Ling Shao. Triple verification network for generalized zero-shot learning. *IEEE Transactions on Image Processing*, 28(1):506–517, **2018**.

[184] Vinay Kumar Verma, Gundeep Arora, Ashish Mishra, and Piyush Rai. Generalized zero-shot learning via synthesized examples. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4281–4289. **2018**.

[185] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(9):2251–2265, **2019**.

[186] Xingxing Zhang, Shupeng Gui, Zhenfeng Zhu, Yao Zhao, and Ji Liu. Atzsl: Defensive zero-shot recognition in the presence of adversaries. *ArXiv*, abs/1910.10994, **2019**.

[187] Morgane Goibert and Elvis Dohmatob. Adversarial Robustness via Label-Smoothing, **2020**. Working paper or preprint.

[188] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037. **2019**.

[189] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems*, pages 5014–5026. **2018**.

[190] Ning Xie, Gabrielle Ras, Marcel van Gerven, and Derek Doran. Explainable deep learning: A field guide for the uninitiated. *arXiv preprint arXiv:2004.14545*, **2020**.

[191] Jingkang Yang, Kaiyang Zhou, Yixuan Li, and Ziwei Liu. Generalized out-of-distribution detection: A survey. *arXiv preprint arXiv:2110.11334*, **2021**.

[192] Ishai Rosenberg, Asaf Shabtai, Yuval Elovici, and Lior Rokach. Adversarial machine learning attacks and defense methods in the cyber security domain. *ACM Computing Surveys (CSUR)*, 54(5):1–36, **2021**.

[193] Yao Deng, Xi Zheng, Tianyi Zhang, Chen Chen, Guannan Lou, and Miryung Kim. An analysis of adversarial attacks and defenses on autonomous driving models. In *2020 IEEE international conference on pervasive computing and communications (PerCom)*, pages 1–10. IEEE, **2020**.

[194] Yuyang Long, Qilong Zhang, Boheng Zeng, Lianli Gao, Xianglong Liu, Jian Zhang, and Jingkuan Song. Frequency domain model augmentation for adversarial attack. In *European Conference on Computer Vision*, pages 549–566. Springer, **2022**.

[195] Zifan Wang, Yilin Yang, Ankit Shrivastava, Varun Rawal, and Zihao Ding. Towards frequency-based explanation for robust cnn. *arXiv preprint arXiv:2005.03141*, **2020**.

[196] Zhe Li, Josue Ortega Caro, Evgenia Rusak, Wieland Brendel, Matthias Bethge, Fabio Anselmi, Ankit B Patel, Andreas S Tolias, and Xaq Pitkow. Robust deep learning object recognition models rely on low frequency information in natural images. *bioRxiv*, **2022**.

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

[198] Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. *arXiv preprint arXiv:2010.01736*, **2020**.

[199] Jisoo Mok, Byunggook Na, Hyeokjun Choe, and Sungroh Yoon. Advrush: Searching for adversarially robust neural architectures. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12322–12332. **2021**.

[200] Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. Wavelet integrated cnns for noise-robust image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7245–7254. **2020**.

[201] Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *Advances in Neural Information Processing Systems*, 32, **2019**.

[202] Alvin Chan, Yew-Soon Ong, and Clement Tan. How does frequency bias affect the robustness of neural image classifiers against common corruption and adversarial perturbations? *arXiv preprint arXiv:2205.04533*, **2022**.

[203] Koki Mukai, Soichiro Kumano, and Toshihiko Yamasaki. Improving robustness to out-of-distribution data by frequency-based augmentation. In *2022 IEEE International Conference on Image Processing (ICIP)*, pages 3116–3120. IEEE, **2022**.

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

[205] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, **2009**.

[206] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. **2011**.

[207] Fisher Yu, Ari Seff, Yinda Zhang, Shuran Song, Thomas Funkhouser, and Jianxiong Xiao. Lsun: Construction of a large-scale image dataset using deep learning with humans in the loop. *arXiv preprint arXiv:1506.03365*, **2015**.

[208] Jihoon Tack, Sangwoo Mo, Jongheon Jeong, and Jinwoo Shin. Csi: Novelty detection via contrastive learning on distributionally shifted instances. *Advances in neural information processing systems*, 33:11839–11852, **2020**.

[209] Dan Hendrycks and Kevin Gimpel. A baseline for detecting misclassified and out-of-distribution examples in neural networks. *arXiv preprint arXiv:1610.02136*, **2016**.

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

[211] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, **2017**.

[212] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032. **2019**.

[213] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in Neural Information Processing Systems*, 33:18661–18673, **2020**.