Explainable robustness for visual classification

Gulshad, S.

Publication date
2022
Document Version
Final published version

Explainable Robustness for Visual Classification – Sadaf Gulshad

Explainable Robustness for Visual Classification

Sadaf Gulshad

Explainable Robustness for Visual Classification

Sadaf Gulshad

This book was typeset by the author using LaTeX 2_ε.

Explainable Robustness for Visual Classification

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het college voor promoties
ingestelde commissie,
in het openbaar te verdedigen in de Agnietenkapel
op woensdag 14 december 2022 te 10:00 uur

door

Sadaf Gulshad

geboren te Rawalpindi, Pakistan

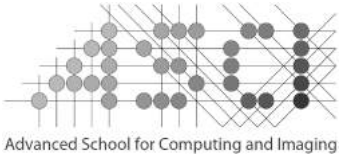
Promotiecommissie

Promotor:	Prof. dr. ir. A.W.M. Smeulders	Universiteit van Amsterdam
Co-promotor:	Dr. H.C. van Hoof	Universiteit van Amsterdam
Overige leden:	Prof. dr. C.G.M. Snoek	Universiteit van Amsterdam
	Prof. dr. T. Gevers	Universiteit van Amsterdam
	Dr. N.J.E. van Noord	Universiteit van Amsterdam
	Dr. J.C. van Gemert	Technische Universiteit Delft
	Prof. dr. Noel E. o'Connor	Dublin City University

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

UNIVERSITEIT VAN AMSTERDAM

The work described in this thesis has been carried out within the graduate school ASCI, dissertation number 442, at the Video and Image Sense lab, and the Delta lab of the University of Amsterdam.



CONTENTS

1

INTRODUCTION

1.1 ROBUSTNESS

In Pakistan it is a right-hand drive, while in the Netherlands it is a left-hand drive. More importantly, the traffic and weather conditions are significantly different. In Pakistan, a driver encounters congested traffic consisting of pedestrians, cars, buses, motorbikes, rikshaws, horse carts, and trucks. On the other hand, although a driver has to be careful about bicycles driving on the same road as cars, traffic is organized in the Netherlands, and rules are strictly followed, as illustrated in Figure 1. Moreover, the central part of Pakistan is medium to warm weather, while in the Netherlands one needs to drive in rain, wind, and snow in the winter. Despite these visually differing circumstances, a person who learns to drive in the Netherlands can quickly adapt to driving in Pakistan. Humans have a remarkable capability when it comes to generalization to unseen visual situations. They easily apply what they have learned from previous experiences to new situations.

Although neural networks show human-level perceptual capabilities [3, 14, 35, 81, 89, 140] under ideal circumstances, they do not show such capabilities for the circumstances they have not encountered in the learning phase. Currently, they require a large amount of data to learn all these different scenarios, otherwise they fail. In one such example in 2018, a self-driving car caused an accident by mistaking a person with an object [10] as in the training of the network of the car jaywalking was never considered. For such

Traffic in Amsterdam, Netherlands

Traffic in Lahore, Pakistan

Figure 1: The image on the left [1] shows a view from Amsterdam, where heavy traffic is organized, while the image on the right [2] shows an example of congested, unorganized traffic in Lahore, Pakistan. Someone who learns to drive in the Netherlands easily adapts to driving in Pakistan. It is the purpose of this thesis to make neural networks robust against such circumstances, at least make them explain why they can not.

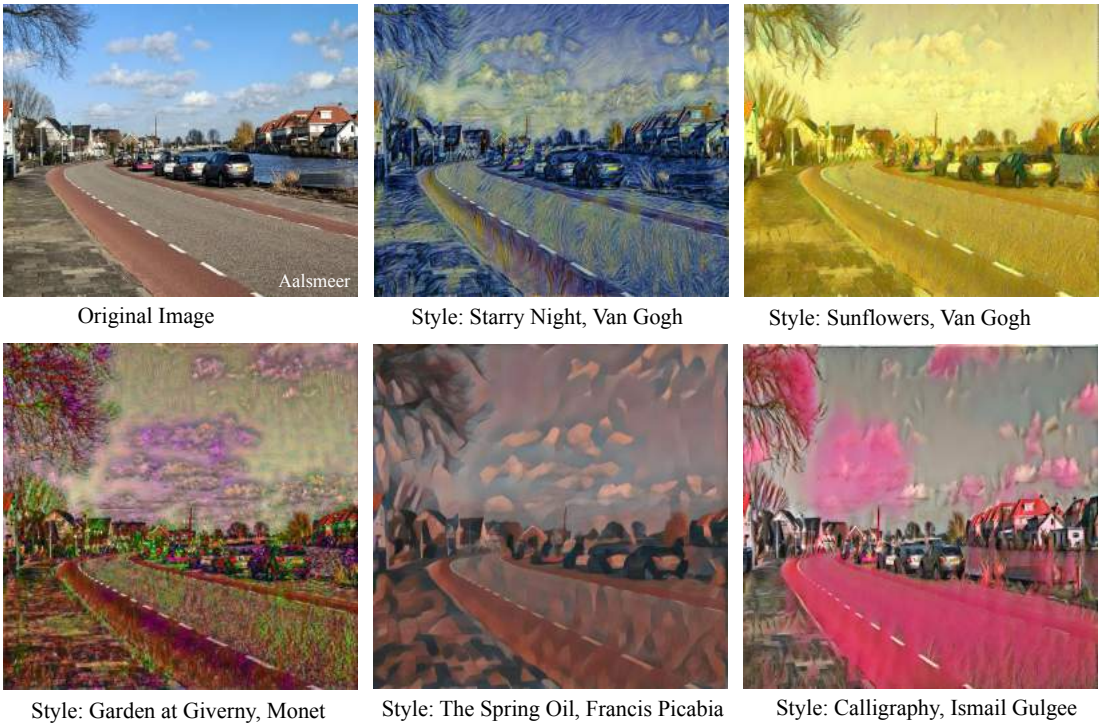


Figure 2: An original image and five differently transformed versions. They are generated using neural style transfer [47] by applying a different style of perturbations for each of the five generated images. Humans can still recognize objects of different styles, but neural networks fail to do so unless trained on such transforms.

reasons, it is important to enhance the robustness of neural networks before deploying them in real-world applications.

Another example in Figure 2 shows the Alsmeerderdijk: the original image and its artificially transformed versions in different painting styles. It is straightforward for a person to recognize the road, the canal, parked cars, boats, trees, clouds, and houses in both the original and the transformed images. Humans easily recognize objects in the presence of perturbations. However, it is challenging for a neural network to recognize objects in perturbed images unless the network is specifically trained on such transformations. Similarly, a self-driving car network exclusively trained on the data collected on a sunny day will fail in the snow or at night, see Figure 3. Hence, in order to make networks intelligent, it is of the utmost importance to robustify them against unseen perturbations. In this thesis, instead of only focusing on seen perturbations, *specific robustness*, we aim to enhance *general robustness*, robustness against perturbations not seen during the training of the network.

Robustness in artificial intelligence is defined as the ability of a system to maintain its performance under the circumstances different from the exact model it was designed for [95]. In this thesis, we do not consider *global robustness* of a task, where the performance of the entire system, say a car with inputs from diverse sensors copes with perturbations. Our focus is *local visual robustness*, the robustness of visual perception.

In this thesis, perturbations are divided into two categories: *natural perturbations* (blur, snow, and alike), see Figure 3 and reference [63], and human-crafted *adversarial perturbations*, see Figure 4, and references [50, 122]. Small, imperceptible, carefully

Figure 3: View from a car's windscreen in different scenarios. Daytime (normal weather), snow, rain and nighttime respectively. A person who learns to drive in one setting can quickly adapt to other scenarios. However, a neural network will likely perform worse, as we will demonstrate in our experiments.

crafted perturbations, used to alter the inputs for fooling deep neural networks, are known as *adversarial examples*, see Figure 4. These adversarial examples push the classifier to the wrong class [122]. Such methods of directed perturbations include the iterative fast gradient sign method [82], the Jacobian-based saliency map attacks [98], one pixel attacks [121], Carlini and Wagner attacks [19] and universal attacks [97]. An example of adversarial perturbations is shown in Figure 4, where the authors showed that for a clean image, the network detects traffic signs correctly. In contrast, for an adversarially perturbed image it fails to detect them, which can cause serious accidents when incorporated into autonomous driving. We make our neural networks robust against natural and adversarial perturbations.

1.2 EXPLAINABLE ROBUSTNESS

Influenced by the interaction of a human with an AI system, [101] defined four properties of explainable AI. 1) Explanation: the reasoning behind the decision. In this thesis, our explanations provide the reasoning and counter-reasoning behind decisions. 2) Meaningful: understandable to the user. We use human-understandable attributes to provide meaningful explanations. 3) Accuracy: accurately reflects the output of the system. We evaluate our explanations qualitatively and quantitatively. 4) Knowledge limits: system only operates under certain conditions and at a certain confidence level. Our explanations, when operated in the perfect scenario of clean images, justify the

Clean Adversarial Clean Adversarial

Figure 4: Traffic light detection for clean and adversarially perturbed inputs [125]. For humans both images still look the same however a neural network fails to detect traffic signs, which can lead to serious consequences. Therefore, we focus on making our classification systems robust against adversarial perturbations.

correct decisions; when operated in imperfect scenarios, provide reasoning behind the failure of the system.

Humans can provide reasons behind their decisions. When in ideal circumstances, decisions are correct, the reasoning supports their decisions. When facing non-ideal circumstances causing incorrect decisions, explanations are used to justify the wrong decision or to understand and correct mistakes. For example, a person drives at the average speed on a clear road, and justifies the speed by saying that the road is clear. However, when a person reduces speed on a snowy road, the reason is that it is slippery. If a person does not reduce speed in the presence of snow, leading to an accident, the next time, the person will know how to act in such a situation, see Figure 5. We provide explanations firstly when the situations are not perfect, i.e., in the presence of perturbations, secondly we make networks robust against perturbations and provide explanations for them, finally we utilize explanations to enhance the robustness i.e., explainably robust.

The most commonly used explainability techniques in recent literature are saliency-based methods [45, 110, 115, 153]. Another group focuses on text-based explanations [59, 99], text-based interpretation with semantic information [33] and generating counterfactual explanations with natural language [60]. Prototype-based explanations have also recently gained attention [22, 34, 108]. However, the techniques mentioned above work in ideal circumstances, they tend to fail in the presence of perturbations. In this thesis, we start from the observation that the explanations are needed more when circumstances deviate from the ideal model. In other words, explainability should come with explainable robustness.

1.3 RESEARCH QUESTIONS

Explainability and robustness, and specifically explainable robustness are essential features required by any neural network before deploying them in real-world applications. Both robustness and explainability are crucial for building trust, providing transparency, and persuading users. They are equally crucial for debugging and analysis on the development side. Therefore, in this thesis, we ask

Figure 5: Humans provide explanations and counter explanations for their decisions. If a person asks why are you driving slow?, then the response given by the other person is because there is ***"snow"*** *on the road, and it is* ***"slippery"****. I will drive* ***"fast"****, if there is* ***"no snow"*** *and it is* ***"not slippery"****. Such explanations and counter explanations when introduced in the neural networks help to understand the decisions, and build trust of users.*

How to make neural network classifiers explainably robust?

To answer this main research question, we dive further into sub questions in each chapter, and begin by asking:

Can an explainability model provide factual (in perfect scenarios) and counterfactual (in imperfect scenarios) explanations?

The question is considered in Chapter 2. Considering our earlier example, when a person "***slows down***" the car, a perfectly understandable reason would be: that there is "***snow***" on the road, and it is "***slippery***". Humans tend to support their decisions by providing counterexamples and counterattributes, such as the speed will be "***high***" if there will be "***no snow***" on the road, and it will "***not be slippery***". Inspired by this style of human explanations, we employ human-understandable visual attributes to provide factual and counterfactual explanations in this chapter.

Factual explanations are provided in the perfect scenarios when the inputs are clean, while counterfactual explanations are provided for imperfect situations, when inputs are perturbed. In order to provide counterexplanations, we use directed perturbations to arrive at the counterclass attribute values. In doing so, we explain what is present and what is absent in the original image. We conduct experiments on both fine-grained and coarse-grained datasets. We verify our attribute-based explanation method quantitatively and qualitatively, and show that attributes provide discriminating and human-understandable explanations for both standard and robust networks.

We demonstrate our attribute-based explanations by providing causal reasoning "because the image contains these attributes, therefore it is classified into this class". Hence, we conclude that attributes provide intuitive factual and in the presence of perturbations counterfactual human understandable explanations, especially for fine-grained classification.

Although perturbations can be utilized to provide counterexplanations, they lead to performance degradation. Therefore, it is essential to make networks robust against perturbations, especially against natural perturbations, which we encounter in real-life scenarios like Gaussian noise, occlusion, blur. After providing human-understandable explanations by utilizing adversarial perturbations, we focus on robustifying networks against adversarial and naturally occurring perturbations and ask

How to integrate natural perturbations in convolutional neural networks for enhancing their general robustness?

The question is considered in Chapter 3. The performance of neural networks is heavily impacted by Gaussian noise or blur in the image [9], [30], [102]. Occlusion or color saturation will also have a similar effect on the network's performance. To counter the effect of these perturbations, we integrate natural perturbations in the network for the purpose of enhancing the robustness of neural networks against perturbations both natural and adversarial, *seen* and *unseen* (during the training).

Previously, [109] trained the network with perturbed images rather than with clean images, or with images perturbed by a learned noise generator [106] to enhance the robustness. [105] proposed to train the network with images from a generative model. In the same category of approaches, it remains an undecided question whether adversarial training [51] is capable of providing robustness to a broad class of natural perturbations [147], [38], [55]. We aim to provide an alternative approach that does not focus on data modification to implement robustness. Instead, we modify a given network by considering transformations to the image filters. We wiggle the weights to implement robustness.

Compared to data augmentation, apart from delivering better results on general robustness, we also have the advantage of providing built-in robustness, where there will be no need to change the data. Our method transforms the network's weights by four different stochastic instantiations of a local elastic transform to cover the local neighborhood by Taylor expansion in the functional space of all classifiers.

Our results show that integrating natural perturbations into the network enhances general robustness. To understand the reasoning behind why wiggling weights with perturbations help against other unseen perturbations, we train a convolutional network on the similarly perturbed images and analyze the results; hence we ask

How to train convolutional neural networks on natural perturbations for enhancing their general robustness?

The research is considered in Chapter 4. We focus on a similar goal as in chapter 3, that is, enhancing the robustness of classifiers against perturbations. Here we propose a training procedure to enhance the robustness, while using similar perturbations as the ones in chapter 3 to understand why integrating perturbations in the network enhances robustness. These tactics of data augmentation are commonly used to enhance the generalization of deep neural networks. [28] showed an improvement in the generalization by randomly occluding parts of images. [148] trained networks on convex combinations of pairs of images and their labels, which led to an improvement in generalization

and robustness against adversarial examples. Similarly, [146] trained on images with regions superimposed from other images. [65] used linear combinations of different data augmentations to enhance generalization. While the methods enhance the generalization of neural networks indeed, most of these methods train networks on non-realistic images, e.g., superimposing parts of two different images. We aim to understand the working of the wiggled weight convolutional network. Therefore, we introduce a training procedure using images with perturbed transforms most similar to the built-in transformations of wiggled weight convolutional networks as possible.

We demonstrate the effectiveness of our natural perturbed training for clean, adversarial, and natural perturbations, both seen and unseen during the training.

Besides robustness, explainability is the key to deploying computer vision models in the real world. Robustness is needed to build trust in the classifier's outcome. Explainability is needed when the circumstances are deviating, so the user can build an understanding of why and when the classifier went off track. Hence, we assert that it is natural to combine robustness and explainability, even at the expense of losing a few percent of the classification accuracy, as one gains trust in return. Therefore, we ask

Can localized visual attributes enhance the general robustness of neural networks, besides providing explanations?

The research question is considered in Chapter 5. Similar to chapter 2, we are inspired by the way humans explain their decisions. Humans discriminate birds by the color of their beak, stripes on their wings, and other attributes, present or absent. In microscopical pathology and radiology, medical students point to visual abnormalities named by their texture. Similarly, here, we explain visual classification by pointing out localized attributes. We propose to learn localized attributes, providing robustness and visual explanation against perturbations in the input. Hence, the aim is to achieve a gain in trust at an acceptable, slight loss in classification accuracy.

We begin by defining attributes as localized and identifying characteristics of an object class. Different from the method in chapter 2 the localized attributes are directly translated into the components of a new transformer architecture. One version of our architecture implements the human-specified class-level attributes as queries to the transformer. The alternative version of our architecture does not use human-specified descriptions, but rather generates localized and identifying attributes itself for our main purpose of providing visually explained robust classification. In contrast, in chapter 2 our model requires human annotated class level attributes for generating explanations. Furthermore, in chapter 2, the purpose of attributes is only to provide explanations for both standard and robust classifiers however here attributes are used to enhance the robustness.

Our attribute-based visual explanations provide us the reasoning behind why a clean input without inflicted perturbations get classified correctly and why the perturbed input get classified into the wrong class.

To summarize, this thesis aims at studying explainable robustness for image classification. We start with providing explanations for a standard neural network-based classifier in ideal (clean input) and non-ideal (perturbed input) situations. Our explanations are also

valid for networks robust against perturbations. Next, we enhance the general robustness of classifiers by integrating the perturbations into the network. Finally, besides providing explanations, we use explanations to enhance explainable robustness. We hope our journey will be able to stimulate more research in the domain of explainability and robustness, and essentially for explainable robustness, as that are the essential components of an intelligent system.

1.4 CO-AUTHORSHIP AND ROLES

For each chapter of the thesis, here we declare the author's contributions

Chapter 2

Gulshad, S. & Smeulders, A.W.M., Counterfactual attribute-based visual explanations for classification. International Conference for Multimedia Retrieval (ICMR), 2021 [53], International Journal of Multimedia Information Retrieval (IJMR), 2021 [54]. *Best paper session at ICMR.*

- S. Gulshad All aspects
- A.W.M. Smeulders Insight, and supervision

Chapter 3

Gulshad, S., Sosnovik, I. & Smeulders, A.W.M., Wiggling Weights to Improve the Robustness of Classifiers. Under submission to ECCV, 2022 [56]

- S. Gulshad All aspects
- I. Sosnovik Theoretical and technical implementation
- A.W.M. Smeulders Insight, and supervision

Chapter 4

Gulshad, S. & Smeulders, A.W.M., Natural Perturbed Training for General Robustness of Neural Network Classifiers. To be submitted to IJCV, 2022 as a combination of chapter 3 and 4 [55]

- S. Gulshad All aspects
- A.W.M. Smeulders Insight, and supervision

Chapter 5

Gulshad, S., Zhao, J., & Smeulders, A.W.M., Learning Localized Attributes for Explainable Robustness of Visual Classifiers. Under submission to ECCV, 2022.

- S. Gulshad All aspects

- J. Zhao Theoretical and technical implementation
- A.W.M. Smeulders Insight, and supervision

2

COUNTERFACTUAL ATTRIBUTE-BASED EXPLANATIONS

2.1 INTRODUCTION

When deploying machine learning and computer vision models in the real world, it is of utmost importance that we explain the decisions made by these models in a human understandable and intuitive way. The preferable procedure to provide such explanations would be as humans explain their decisions. For example, when a person classifies a bird into the "***Cardinal***" class, the reason provided by the person is: because it has a "***Crested head***" and a "***Red beak***". Humans also tend to support their decisions by providing counterexamples and counterattributes such as, this bird would be classified into the class "***Pine Grosbeak***" if it will have a "***Plain head***" and a "***Black beak***" as shown in Figure 6. Inspired by human explanations, in this paper we employ human understandable visual attributes for providing factual and counterfactual explanations.

A large body of work in explainable AI focuses on explaining the decisions of neural network-based classifiers using saliency maps [115,153]. Saliency maps highlight the part of the image which supports the classification however, the support to the classification might be distributed across the whole image, or might lie in the color or texture of the object. Hence, it becomes difficult to localize the part of the image responsible for the classification, especially for fine-grained datasets. Furthermore, saliency maps tell us about what is present in the image and do not provide any information about what is absent, i.e. counterfactual information. Therefore, in this work, we focus on human nameable attributes for providing the reasoning behind specific decisions and perturbations to arrive at attributes belonging to counterclasses to provide counterfactual explanations.

In a closely related work [52], the authors provided counterfactual explanations for classification decisions by replacing the part of the original image with the similar part from the distractor image belonging to the counterclass, such that the class of the image changes. However, their method is pixel-based, hence requires matching imaging conditions such as pose and illumination. In contrast, in this work, we introduce perturbations in the images so that the attribute values change to the counterclass attribute values.

In a recent work for the different purpose of enhancing the generalization power of visual question answering systems [4] authors utilized counterfactuals and trained the network with counterexamples. Similarly, in our work, we improve the generalization and robustness of the neural network-based classifier by training it with counterexamples. However, we go a step further and provide counterfactual explanations for this network.

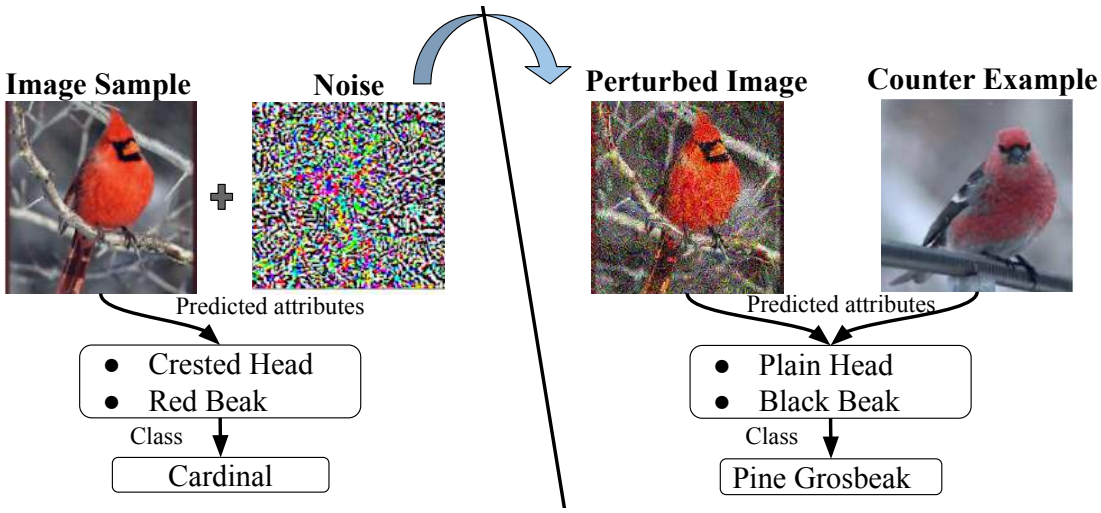


Figure 6: We use attributes to explain why an image on the left is classified into the Cardinal class rather than the Pine Grosbeak class on the right. And we use attributes with examples to explain when it will be classified as a Pine Grosbeak by exploiting perturbed examples and their attribute values. We show that when the predicted attributes for the image change from "Crested Head" and "Red Beak" to "Plain Head" and "Black Beak", the image will be classified as Pine Grosbeak.

We define the closeness of classes in the embedding space based on the attribute similarity and evaluate our method when images get misclassified into the closer counterclass [122] as well as when we force them to be misclassified into a distant counterclass [20]. We complement our attribute-based explanations with counterexample-based explanations by selecting the examples containing counterattributes.

Our main contributions are given as follows:

- We provide novel explanations for classification decisions by utilizing intuitive factual and counterfactual attributes and examples.
- We study the change in attribute values when images are perturbed to provide counterfactual explanations from any alternative counterclass as well as when images are perturbed to provide counterfactual explanations from our desired counterclass.
- We propose a novel method to assist our attribute-based explanations with counterexamples, selected based on these counterattributes.

We evaluate our attribute-based explanations *quantitatively* and *qualitatively* for a *standard* as well a *robust* network. Our results on three different datasets of varying sizes and granularity show that attributes provide effective factual and counterfactual explanations for classifier decisions. This paper is an extended version of our conference paper [53].

2.2 RELATED WORK

Explaining the output of a decision maker is commonly motivated by the need to build user trust before deploying them into a real world environment [37, 57, 90].

2.2.1 *Explainability*

Previous work for visual classification explanation is broadly grouped into two types: 1) *rationalization*, that is, justifying the network's behavior and 2) *introspective explanation*, that is, showing the causal relationship between input and the specific output [36]. The first group has the benefit of being human understandable, but it lacks a causal relationship between input and output. The second group incorporates the internal behavior of the network, but lacks human understandability. In this work, we explain the decisions of neural networks in the human style of explanations by singling out specific attributes for positive evidence when the image is classified correctly and by following specific attributes for negative evidence when the image is directed for misclassification in a counterclass.

An important group of work on understandibility focuses on text-based class discriminative explanations [59, 99], text-based interpretation with semantic information [33] and generating counterfactual explanations with natural language [60], they all fall in the *rationalization* category. Text-based explanations are orthogonal to our attribute-based explanations, as attributes tend to deliver the key-words in the sentence and carry the quintessence for the semantic distinction. Especially for fine-grained classification, all sentences for all classes tend to display the same structure hence, the core of the semantic distinction between classes lies in attributes where we put our emphasis. Generating sentences is valuable, but largely orthogonal to our approach.

To tackle the similar task of explaining visual decisions, there is the large body of work on activation maximization [115, 153], learning the perturbation mask [45], learning a model locally around its prediction, and finding important features by propagating activation differences [104, 112]. They all fall in the group of *introspective explanations*. All these approaches use saliency maps for explanation. We observe that saliency maps [110] are frequently weak in justifying classification decisions, especially for fine-grained images. For instance, in Figure 7 the saliency map of a clean image classified into the ground truth class, "red-winged blackbird", and the saliency map of a misclassified perturbed image, look quite similar. Instead, by grounding the predicted attributes, one may infer that the "orange wing" is important for "red-winged blackbird" while the "red head" is important for "red-faced cormorant". Indeed, when the attribute value for orange wing decreases and for red head increases, the image gets misclassified. Therefore, we propose to predict and ground attributes for both clean and perturbed images to provide visual as well as attribute-based interpretations.

Counterfactual Explanations. Explanations which consider counterdecisions or counteroutcomes are known as *counterfactual explanations* [87]. An interesting approach in a recent paper [52] proposes to generate counterfactual explanations by selecting a distractor image from a counterclass and replacing the region in the input image with a region from the distractor image, such that the class of the input image changes into the class of the distractor image. Pixel-based replacements pose high restrictions on the similarity of viewpoint, pose and scene between the two images, which makes the selection and replacement of the patches difficult. We follow the same inspiration of human-motivated counterexamples. However, our approach focuses on attributes for generating explanations, as they contain the semantic core of the distinction between two competing classes and, attributes can naturally incorporate large changes in imaging

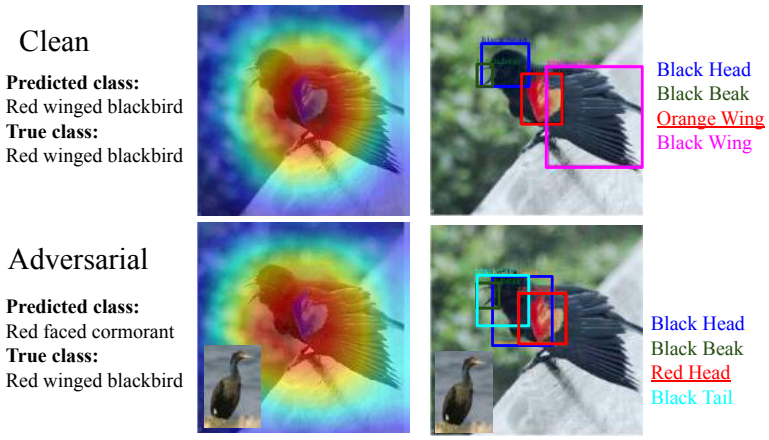


Figure 7: Fine-grained images are difficult to explain with saliency maps: when the answer is wrong, often saliency-based methods (left) fail to detect what went wrong. Instead, attributes (right) provide intuitive and effective visual and textual explanations.

conditions of size, illumination and viewpoint. Additionally, we use perturbations to change the class of the input image, we analyze which attributes lead to the change in class.

Another closely related work, [75], focuses on the multimodal complementarity of text and image for explanations. They maximize the interaction information between class predictor and explanation generator by simultaneously training them using variational lower bound. However, by the nature of their method their example-based explanations will be visually completely different from the input image. In our work, by using the method of directed perturbations and discriminating attributes, we are capable of selecting the most critical counterexamples as the most effective explanations.

In [4], authors utilized counterfactuals for enhancing the generalization and applicability of visual question answering systems. However, in our work for providing explanations, we increase the generalization and robustness of neural network classifier by training it on counterfactuals. After robustification we verify our method on the robustified network by studying the change in attributes for it.

2.2.2 *Adversarial Examples*

Untargeted Methods. Small, carefully crafted perturbations, called *adversarial perturbations*, have been used to alter the inputs of deep neural networks, which results in *adversarial examples*. These adversarial examples drive the classifiers to the wrong class [122]. Such methods of directed perturbations include iterative fast gradient sign method (IFGSM) [82], the Jacobian-based saliency map attacks [98], one pixel attacks [121], Carlini and Wagner attacks [19] and universal attacks [97]. Here, our aim is to utilize the directed noise from adversarial examples to study the change in attribute values. Therefore, we select the IFGSM-method which is fast and strong for our experiments to lead images into counterclasses.

Targeted Methods. When small adversarial perturbations are introduced in the images to misclassify them into the desired counter classes, are called *targeted attacks* [21]. Targeted attacks are stronger and more difficult to achieve than *untargeted attacks* because

the algorithm needs to find the perturbations, which will misclassify the image into the desired class instead of misclassification into any alternative class [20] i.e. *untargeted attacks*. Besides studying the change in attribute values for untargeted attacks, here we also study the change in the attribute values when images are directed into desired classes. For this purpose, we utilize the targeted version of IFGSM method and compare the results for untargeted and targeted attacks to verify whether our proposed attribute-based counterfactual explanations also function for targeted attacks.

2.2.3 *Adversarial Examples for Explainability.*

Adversarial examples have been used for understanding neural networks. [68] aims at utilizing adversarial examples for understanding deep neural networks by extracting the features which provide the support for classification into the target class. In this paper, instead of providing feature based visualizations, we focus on human understandable attributes for providing explanations for decisions. In [72], the authors proposed a data-path visualization module consisting of the layer level, the feature level, and the neuronal level visualizations of the network for clean as well as for adversarial images. In contrast, we focus on exploiting adversarial examples to generate intuitive factual and counterfactual human understandable explanations with attributes and visual examples.

In [150], the authors investigated adversarially trained robust convolutional neural networks by constructing input images with different textual transformations while at the same time preserving the shape information. They do this to verify the shape bias in adversarially trained networks compared with standard networks. Similarly, in [126], the authors showed that saliency maps from adversarially trained robust networks align well with human perception. In our work, we also provide explanations when an image is correctly classified with an adversarially trained robust network and verify that the attributes predicted by our method with a robust network still retain their discriminative power for explanations.

Adversarial Examples and Counterfactual Explanations In a closely related work [69] authors reveal the duality relationship between adversarial examples and explanations. They argue that adversarial examples could be generated from counterexamples, and counterexamples could be generated from adversarial examples. We follow a similar idea, but instead propose to utilize adversarial examples for explanations in the presence of human understandable attributes.

Similarly, [16] tries to solve the paradox that previous research [129] shows that adversarial examples and counterfactual explanations are equivalent, then where lies the difference between them? They argue that this paradox could be solved by properly studying the semantics (i.e. neuronal activations) of counterfactuals for providing explanations. In this paper, instead of focusing on solving the paradox between adversarial examples and counterfactual explanations, we make use of adversarial examples with attributes to provide counterfactual explanations.

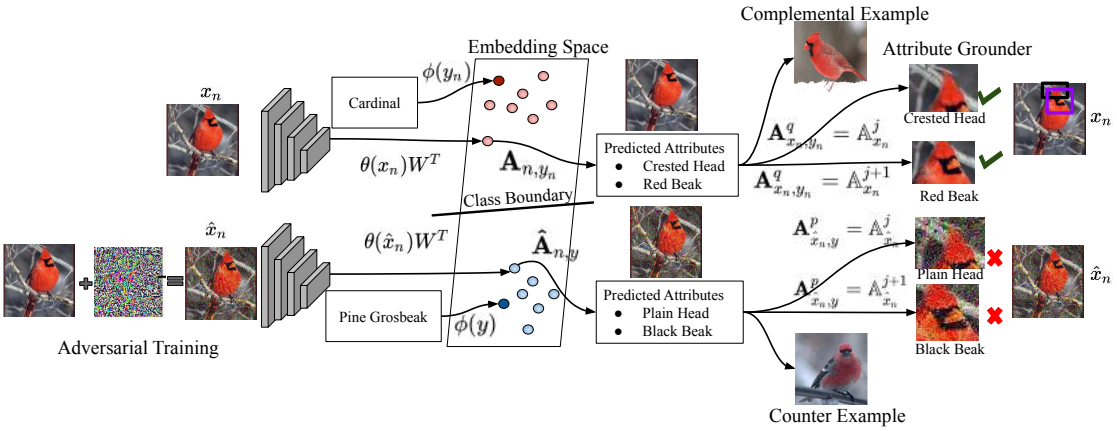


Figure 8: Interpretable attribute prediction-grounding model. After an adversarial training step, image features of both clean $\theta(x_n)$ and adversarial images $\theta(\hat{x})$ are extracted using Resnet and mapped into attribute space $\phi(y)$ by learning the compatibility function $F(x_n, y_n; W)$ between image features and class attributes. Attributes predicted by attribute-based classifier $\mathbf{A}^q_{x_n,y_n}$ are grounded by matching them with attributes predicted by Faster-RCNN $\mathbb{A}^j_{x_n}$ for clean and adversarial images. Examples are selected based on attribute similarity between adversarial image and adversarial class images for visual explanations. Hence, clean image attributes lead to complemental explanations while, adversarial image attributes lead to counterfactual explanations.

2.3 METHOD

2.3.1 *Adversarial Perturbations*

Given n-th image x_n and its respective ground truth class y_n predicted by a classifier $f(x_n)$, an image $\hat{x}_n$ is generated by adding adversarial perturbations to it such that the classifier $f(\hat{x}_n)$ predicts y, where $y_n \neq y$, and x_n and $\hat{x}_n$ are close according to some distance metric. Next, we present the method for generating adversarial examples through *untargeted attacks* [20] and *targeted attacks* [20] [21].

Untargeted Attacks. We leverage IFGSM method [82] to generate adversarial perturbations. The mechanism for generating adversarial examples through basic iterative method is given by:

$$\begin{aligned} \hat{x}_n^0 &= x_n \\ \hat{x}_n^{i+1} &= \text{Clip}_\epsilon\{\hat{x}_n^i + \alpha \text{Sign}(\nabla_{\hat{x}_n^i} \mathcal{L}(\hat{x}_n^i, y_n))\} \end{aligned} \tag{2.1}$$

where, $\hat{x}_n^0$ is the input image at step $i = 0$, $\nabla_{\hat{x}_n^i} \mathcal{L}$ is the derivative of the loss function w.r.t to the current input image, α is the step size taken at step i in the direction of sign of the gradient, and finally the result is clipped by Clip_ϵ.

Targeted Attacks. For targeted attacks, we target our input image to be misclassified into a specific class y_t. The following equations are used to create adversarial perturbations for misclassification in the target class.

$$\hat{x}_n^0 = x_n$$
$$\hat{x}_n^{i+1} = \text{Clip}_\epsilon\{\hat{x}_n^i - \alpha\text{Sign}(\nabla_{\hat{x}_n^i}\mathcal{L}(\hat{x}_n^i, y_t))\} \tag{2.2}$$

In the targeted attacks, we maximize the loss against ground truth class y_n and minimize the loss against target class y_t.

2.3.2 *Adversarial Robustness*

Adversarial Training. Adversarial training [122] is one of the state-of-the-art method for robustness against adversarial perturbations. In adversarial training, the model $f^r(\hat{x}_n)$ finds the worst case adversarial examples and trains the network on these adversarial examples besides training it on clean images to make it robust against adversarial perturbations. Hence, this leads to an improvement in performance against adversarial perturbations. The following objective function is minimized in adversarial training:

$$\mathcal{L}_{adv}(x_n, y_n) = \gamma\mathcal{L}(x_n, y_n) + (1-\gamma)\mathcal{L}(\hat{x}_n, y) \tag{2.3}$$

where, $\mathcal{L}(x_n, y_n)$ is the classification loss for clean images, $\mathcal{L}(\hat{x}_n, y)$ is the loss for adversarial images and γ regulates the loss to be minimized.

2.3.3 *Attribute Prediction*

We use class attributes available with the dataset to predict per image attributes and provide explanations for classification. The model is shown in the Figure 8. At training time, our network learns to map image features closer to their ground truth class attributes and farther from other classes in the embedding space. During test time when clean image features are projected in the learned embedding space the image gets mapped closer to the ground truth class attributes e.g. "Crested head" and "Red beak" associated with the ground truth class "Cardinal", see Figure 8. However, an adversarially perturbed image gets mapped closer to the wrong class attributes e.g. "Plain head" and "Black beak" belonging to the counterclass 'Pine Grosbeak", Figure 8.

Given the n-th input image features $\theta(x_n) \in \mathcal{X}$ and output class attributes $\phi(y_n) \in \mathcal{Y}$ from the sample set, $\mathcal{S} = \{\theta(x_n), \phi(y_n), n = 1....N\}$ we employ SJE [6] to predict attributes in an image. SJE learns to map $: \mathcal{X} \longrightarrow \mathcal{Y}$ by minimizing the empirical risk of the form $\frac{1}{N}\sum_{n=1}^{N}\Delta(y_n, (x_n))$, where $\Delta : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ estimates the cost of predicting (x_n) when the ground truth label is y_n.

A compatibility function $F : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is defined between input $\mathcal{X}$ and output $\mathcal{Y}$ space:

$$F(x_n, y_n; W) = \theta(x_n)^T W\phi(y_n) \tag{2.4}$$

Pairwise ranking loss $\mathbb{L}(x_n, y_n, y)$ is used to learn the parameters (W):

$$\Delta(y_n, y) + \theta(x_n)^T W \phi(y_n) - \theta(x_n)^T W \phi(y) \tag{2.5}$$

At test time, attributes are predicted for clean images by projecting image features on to the learned embedding space. It is given by:

$$\mathbf{A}_{n,y_n} = \theta(x_n) W \tag{2.6}$$

and for adversarial images, by:

$$\hat{\mathbf{A}}_{n,y} = \theta(\hat{x}_n) W \tag{2.7}$$

The image is assigned the label of the nearest output class attributes $\phi(y_n)$.

2.3.4 *Attribute Grounding*

Thereafter, we ground the predicted attributes on the images for better visual explanations using a pre-trained Faster-RCNN as in [8]. The pre-trained Faster-RCNN $\mathcal{F}(x_n)$ model predicts bounding boxes b^j. For each bounding box j in each image, x_n it predicts a class $\mathbb{Y}^j_{x_n}$ and an attribute $\mathbb{A}^j_{x_n}$ [7].

$$b^j_{x_n}, \mathbb{A}^j_{x_n}, \mathbb{Y}^j_{x_n} = \mathcal{F}(x_n) \tag{2.8}$$

where j is the bounding box index.

Attribute Selection for Grounding. As all the attributes predicted for an image can not be visualized due to visual constraints. Therefore, we select the most discriminative attributes for grounding on the images. Attributes are selected based on the criterion that they change the most when the image is perturbed with the adversarial noise. For clean images we use:

$$q = \underset{i}{\operatorname{argmax}}(\mathbf{A}^i_{n,y_n} - \phi(y^i)) \tag{2.9}$$

and for adversarial images we use:

$$p = \underset{i}{\operatorname{argmax}}(\hat{\mathbf{A}}^i_{n,y} - \phi(y^i_n)). \tag{2.10}$$

where i is the attribute index, $\mathbf{A}^i_{n,y_n}$ and $\hat{\mathbf{A}}^i_{n,y}$ are attributes predicted by SJE for clean and adversarial images respectively. $\phi(y^i)$, $\phi(y^i_n)$ indicate the counterclass and ground truth class attributes, respectively. q and p are indexes of the most discriminative attributes selected based on our criterion.

After selecting the most discriminative attributes predicted by SJE using equation 2.9 and 2.10, we search for the selected attributes $\mathbf{A}^q_{x_n,y_n}, \mathbf{A}^p_{\hat{x}_n,y}$ in the attributes predicted by RCNN for each bounding box $\mathbb{A}^j_{x_n}, \mathbb{A}^j_{\hat{x}_n}$. When the attributes predicted by SJE and Faster-RCNN are matched, that is $\mathbf{A}^q_{x_n,y_n} = \mathbb{A}^j_{x_n}$, $\mathbf{A}^p_{\hat{x}_n,y} = \mathbb{A}^j_{\hat{x}_n}$ we ground them on their respective clean and adversarial images. As shown in the Figure 8, the attributes "Crested head" and "Red beak" are grounded on the image, while "Plain head" and "Black beak"

could not be grounded because there is no visual evidence present in the image for these attributes.

2.3.5 *Example-based Explanations*

Besides providing attribute-based explanations, we propose to provide counterexample-based explanations, as shown in the Figure 8. We compare the results for example-based explanations by selecting examples randomly from the counterclass with examples selected based on attributes Figure 18.

Example Selection through Attributes. The procedure for example-based explanations using attributes is detailed in the Algorithm 1 and the results are shown in Figure 17 and Figure 18. Given clean images classified correctly, and adversarial images misclassified and their predicted attributes, we search for attributes in the adversarial class which are most similar to the attributes of the adversarial image and select these images as counterexamples i.e. a "Pine Grosbeak" image with the attributes "Plain head" and "Black beak" is selected as a counterexample Figure 8.

Algorithm 1 Example Selection through Attributes

1: Given adversarial images $\hat{x}_{n,y}$, clean images x_{n,y_n}, adversarial image attributes $\hat{\mathbf{A}}_{n,y}$, clean image attributes $\mathbf{A}_{n,y_n}$, adversarial classes y
2: **for** each adversarial image $\hat{x}_{n,y}$ **do**
3: Select all the images from adversarial class $x_{n,y}$
4: **for** each image in adversarial class $x_{n,y}$ **do**
5: $s = \underset{i}{\operatorname{argmin}} \parallel \hat{\mathbf{A}}^i_{n,y} - \mathbf{A}^i_{n,y} \parallel_2$
6: **end for**
7: **end for**
8: **return** Selected examples from adversarial class $x^s_{n,y}$

2.3.6 *Attribute Analysis Method*

Finally, in this section, we introduce our techniques for quantitative analysis on the predicted attributes.

Predicted Attribute Analysis: Standard Network. In order to perform analysis on attributes in embedding space, we consider the images which are correctly classified without perturbations and misclassified with perturbations. Our aim is to analyze the change in attributes in embedding space to verify that attributes change with the change in the class.

We contrast the Euclidean distance between predicted attributes of clean and adversarial samples:

$$d_1 = d\{\mathbf{A}_{n,y_n}, \hat{\mathbf{A}}_{n,y}\} = \parallel \mathbf{A}_{n,y_n} - \hat{\mathbf{A}}_{n,y} \parallel_2 \tag{2.11}$$

with the Euclidean distance between the ground truth attribute vector of the correct and adversarial classes:

$$d_2 = d\{\phi(y_n), \phi(y)\} = \parallel \phi(y_n) - \phi(y)) \parallel_2 \tag{2.12}$$

where, $\mathbf{A}_{n,y_n}$ denotes the predicted attributes for the clean images classified correctly, and $\hat{\mathbf{A}}_{n,y}$ denotes the predicted attributes for the adversarial images misclassified with a standard network. The correct ground truth class attribute are referred to as $\phi(y_n)$ and adversarial class attributes are referred to as $\phi(y)$.

Predicted Attribute Analysis: Robust Network. We compare the distances between predicted attributes of only adversarial images that are classified correctly with the help of an adversarially robust network $\hat{\mathbf{A}}^r_{n,y_n}$ and classified incorrectly with a standard network $\hat{\mathbf{A}}_{n,y}$:

$$d_1 = d\{\hat{\mathbf{A}}^r_{n,y_n}, \hat{\mathbf{A}}_{n,y}\} = \| \hat{\mathbf{A}}^r_{n,y_n} - \hat{\mathbf{A}}_{n,y} \|_2 \tag{2.13}$$

with the distances between the ground truth class attributes $\phi(y_n)$ and ground truth adversarial class attributes $\phi(y)$:

$$d_2 = d\{\phi(y_n), \phi(y)\} = \| \phi(y_n) - \phi(y)) \|_2 \tag{2.14}$$

2.3.7 *Implementation Details*

Image Features and Adversarial Examples. We extract image features and generate adversarial images using the fine-tuned Resnet-152. Adversarial attacks are performed using the basic iterative method with epsilon ϵ values 0.01, 0.06 and 0.12. The l_∞ norm is used as a similarity measure between clean input and the generated adversarial example. In order to generate adversarial examples for untargeted attacks, the algorithm perturbs the images such that they get misclassified into any alternative counter class. In order to generate adversarial examples for targeted attacks, we direct adversarial examples to be misclassified into randomly selected classes.

Adversarial Training. As for adversarial training, we repeatedly computed the adversarial examples while training the fine-tuned Resnet-152 to minimize the loss on these examples. We generated adversarial examples using the projected gradient descent method. This is a multistep variant of FGSM with epsilon ϵ values 0.01, 0.06 and 0.12 respectively for adversarial training as in [91].

Attribute Prediction and Grounding. At test time, the image features are projected onto the learned attribute space and attributes per image are predicted. The image is assigned with the label of the nearest ground truth attribute vector. Since we do not have ground truth part bounding boxes for any of the attribute datasets, the predicted attributes are grounded by using Faster-RCNN pre-trained on the Visual Genome Dataset [79].

2.4 EXPERIMENTS AND RESULTS

2.4.1 *Datasets*

We experiment on three datasets, Animals with Attributes 2 (AwA) [83], Large attribute (LAD) [151] and Caltech UCSD Birds (CUB) [103]. AwA contains, 37322 images (22206 training / 5599 validation / 9517 test) with 50 classes and 85 attributes per class. LAD has, 78017 images (40957 training / 13653 validation / 23407 test) with 230 classes and 359 attributes per class. CUB consists of 11,788 images (5395 training / 599 validation / 5794 test) assigned to 200 fine-grained categories of birds with 312 attributes

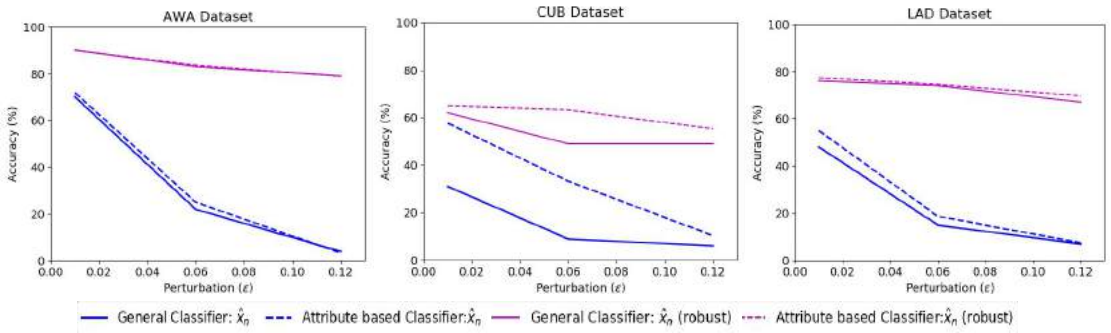


Figure 9: Untargeted Attacks: Comparing the accuracy of the general classifier and the attribute-based classifier for adversarial examples generated with untargeted attacks to investigate the change in attributes. We evaluate both classifiers by extracting features from a standard network and the adversarially robust network. The drop in the performance with the increase in the level of perturbations shows that the attributes start pointing towards the counter classes (blue curves). The improvement in the performance with robustification shows that with an adversarially robustified network, the attributes again start pointing towards the ground truth class (purple curves).

per class. All three datasets contain real-valued class attributes representing the degree of presence of an attribute in a class. For the qualitative analysis with grounding, we select 50 attributes that change their value most for the CUB, 50 attributes for AWA, and 100 attributes for the LAD dataset. They are selected by equation 2.9 and 2.10, since it is difficult for humans to understand all the attributes grounded on the images.

The Visual Genome Dataset [79] is used to train the Faster-RCNN model, which extracts the bounding boxes using 1600 object and 400 attribute annotations. Each bounding box is associated with an attribute and the class, e.g. a brown bird.

2.4.2 *Comparing General and Attribute-based Classifiers*

In the first experiment, we compare the general classifier $f(x_n)$ and the attribute-based classifier (x_n) in terms of the classification accuracy on clean images to see whether the attribute-based classifier performs equally well.

We find that, the attribute-based and general classifier accuracies are comparable for AWA (general: 93.53, attribute-based: 93.83). The attribute-based classifier accuracy is slightly higher for LAD (general: 80.00, attribute-based: 82.77), and lower for CUB (general: 81.19, attribute-based: 76.90) dataset. The overall impression is that both general and attribute-based classifiers perform equally well.

2.4.3 *Attribute-based Explanations: Standard Network*

In the second experiment we study the change in attributes with a standard network to demonstrate that by introducing perturbations in the images the attribute values change such that the class of the image changes to the counterclass and hence provide intuitive counterexplanations. We study the change in attribute values both when the counterclass

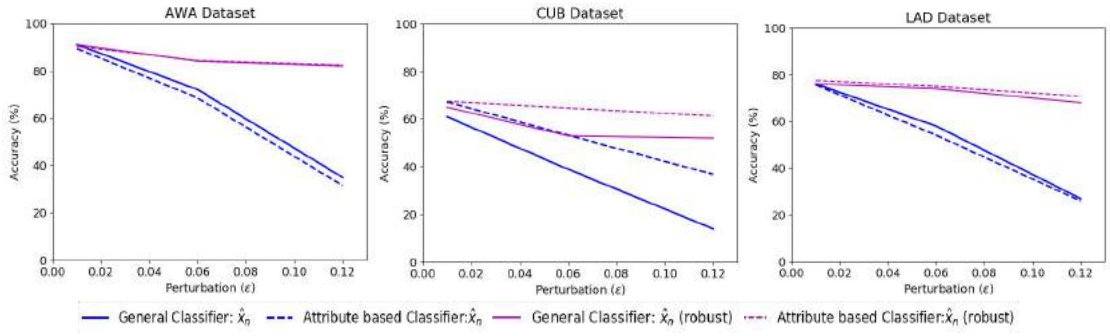


Figure 10: Targeted Attacks: Comparing the accuracy of the general classifier and the attribute-based classifier for adversarial examples generated with targeted attacks to investigate the change in attributes. We evaluate both classifiers by extracting features from a standard network and the adversarially robust network. The drop in the performance with the increase in the level of perturbations shows that the attributes start pointing towards the counter classes (blue curves). However, the drop is not significant when compared to untargeted attacks. Similarly, with the adversarial robustness the performance improves, and the attributes start pointing towards the ground truth class, however the improvement is also not as significant as for the untargeted attacks (purple curves).

is any other class i.e. untargeted, and when we direct the image into a specific class i.e. targeted.

By Performing Classification based on Attributes

Untargeted Attacks. With untargeted adversarial attacks, the accuracy of both the general and attribute-based classifiers drops with the increase in perturbations, see Figure 9 (blue curves). The drop in accuracy of the general classifier for the fine-grained CUB-dataset is higher than the coarse-grained AWA dataset. For example, at $\epsilon = 0.01$ for the CUB dataset the general classifier's accuracy drops from 81% to 31% , while for the AWA dataset it drops from 93% to 70% and for the LAD dataset it drops from 80% to 50%. However, compared to the general classifier, the drop in accuracy with the attribute-based classifier for CUB dataset is less $\approx$ 20%. For the coarse-grained datasets AWA and LAD, the drop is almost the same for both attribute-based and general classifiers. The limited drop in accuracy for the CUB dataset with the attribute-based classifier when compared to the general classifier, is attributed to the fact that for fine-grained datasets there are many attributes common among classes. Therefore, in order to misclassify an image, a significant number of attributes need to change their values. For a coarse-grained dataset, changing a few attributes is sufficient for misclassification. Overall, the drop in the accuracy due to the perturbation demonstrates that the attribute values change towards those that belong to the new class. Hence, attributes explain the misclassifications into the counterclasses well. This also concludes that attributes contain the crucial characteristics for discrimination between classes.

Targeted Attacks. In the untargeted attacks, the algorithm misclassifies the image into any alternative class which could be a closer class, i.e. a class with the majority of attribute values same as the ones from the ground truth class. In contrast, targeted

adversarial attacks force the image to be misclassified into a randomly selected desired class which could be far away from the ground truth class i.e. the attribute values between both classes differ significantly, hence making the targeting into this class difficult. We evaluate our method for misclassification into a closer class as well as for a distant class.

The accuracy of both general and attribute-based classifiers drop with the increase in perturbations, see Figure 10 (blue curves). However, compared to the drop in performance with untargeted attacks the drop with targeted attacks is lower, see Figure 9 and Figure 10 (blue curves). This is due to the fact that in untargeted attacks the images are misclassified into closer classes, while with the targeted attacks images get misclassified into distant classes.

By contrasting the drop in the accuracy of the general classifier between three datasets using targeted attacks we observe that the fine-grained CUB-dataset leads to a higher drop in the performance as compared to the AWA, and LAD datasets Figure 10 (blue solid curves). Although the drop with targeted attacks is lower than untargeted attacks, but the overall behavior in the drop is the same for both untargeted and targeted attacks. For instance, at $\epsilon = 0.06$ the accuracy drops from 81% to 39% for CUB-dataset, while for AWA dataset it drops from 93% to 72% and for LAD dataset it drops from 80% to 58%. While the drop in the accuracy with the attribute-based classifier for CUB-dataset reduced to almost half i.e. $\approx 23\%$ and increased for AWA and LAD dataset i.e. $\approx 25\%$ and $\approx 29\%$ respectively. Similar to the general classifier, attribute based classifier for targeted attacks also shows the same behavior as attribute based-classifier for untargeted attacks. Hence, this further supports our argument that for fine-grained datasets as there are numerous attributes common among the classes therefore we need to change many of them in order to change the class and provide explanations based on the attributes. While, for the coarse grained datasets, only by changing a few attributes we can cause misclassification and explain it.

Overall, the lack in the drop of performance for an attribute based classifier with the targeted attacks as compared to untargeted attacks shows that the change in attribute values towards the counterclass is less significant with the targeted attacks. Hence, attribute values with untargeted attacks provide better counterexplanations than attribute values with the targeted attacks.

By Computing Distances in the Embedding Space

We contrast the Euclidean distance between predicted attributes of clean and adversarial samples using equation 2.11 and 2.12. The results are shown in Figure 11. We observe that for the AWA dataset, the distances between the predicted attributes for adversarial and clean images d_1 are smaller than the distances between the ground truth attributes of the respective classes d_2. The closeness in predicted attributes for clean and adversarial images as compared to their ground truths shows that attribute values change towards the wrong class but not completely. This is due to the fact that for coarse classes, only a small change in attribute values is sufficient to change the class.

The fine-grained CUB-dataset behaves differently. The overlap between d_1 and d_2 distributions demonstrates that attributes of images belonging to fine-grained classes change significantly as compared to images from coarse categories. As the fine-grained classes are closer to one another and many attributes are common among fine-grained

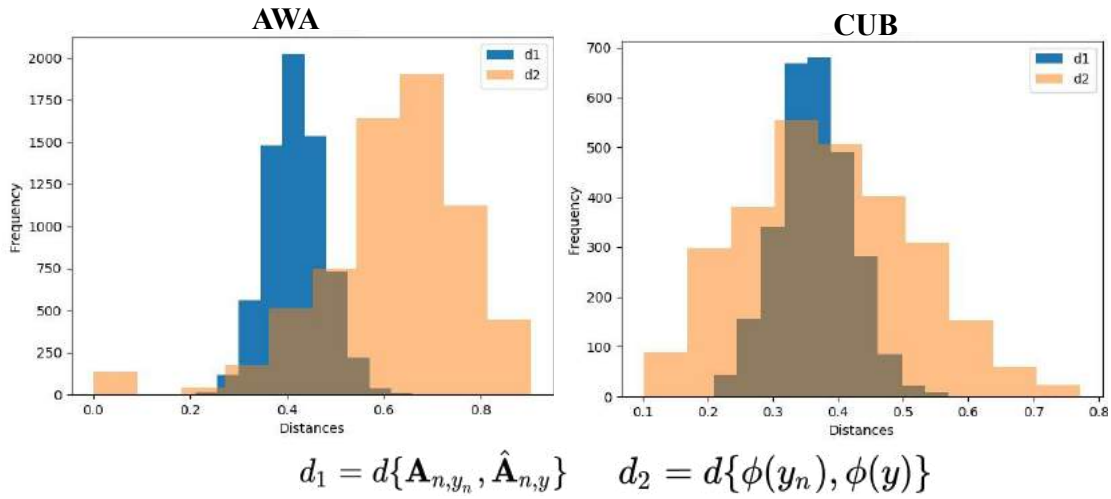


Figure 11: Attribute value distance plots for clean and adversarial images with a standard network. The complete overlap for the CUB-dataset shows that fine-grained datasets require change in significant no of attribute to change the class. While the small overlap for the coarse-grained AWA dataset shows that the change in a few attributes is sufficient to change the class.

classes. Thus, it requires changing the attributes significantly to cause misclassification. Hence, for the coarse-grained dataset, the attributes change minimally, while for the fine-grained dataset they change significantly.

Qualitative Analysis

Untargeted Attacks. We observe in Figure 13 that the most discriminative attributes for the clean images are coherent with the ground truth class however, for adversarial images they are coherent with the wrong class, thus explaining the wrong class. For example "red head, black wing, black eye" attributes are responsible for the classification of clean image into correct class and when the value of "red head" attribute decreases and "grey beak, white underparts" increases the image gets misclassified into wrong class. Figure 12 reveals the results for the groundings on perturbed images. The attributes which are not related to the correct class, the ones that are related to the counterclass can not get grounded or get grounded at the wrong spots in the image as there is no visual evidence that supports the presence of these attributes. For example, "black tail" is related to the counterclass and is not present in the adversarial image. Hence, black tail" got wrongly grounded. This indicates that attributes for the clean images correspond to the ground truth class and for adversarial images correspond to the counterclass. Additionally, only those attributes common among both the counterand the ground truth classes get grounded on adversarial images.

Hence, our method provides explanations for both fine and coarse-grained classifications when the images get misclassified into similar classes or dissimilar classes.

Targeted Attacks. Figure 14 reveals the results for grounding the attributes when the images are misclassified with targeted attacks. As in the targeted attacks we direct images into random classes, we observe that images get misclassified into visually dissimilar classes. The attributes predicted for perturbed images also correspond to

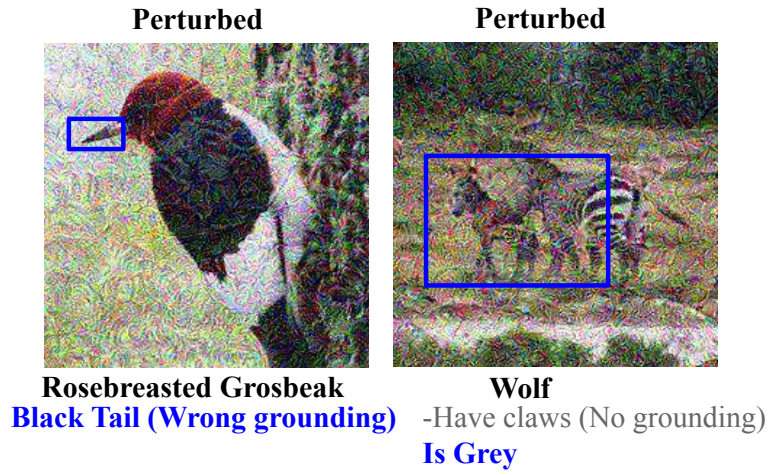


Figure 12: Explanation of a wrong classification due to wrong or missing attribute grounding. For perturbed images, attributes either get grounded on wrong spots or are missing because their visual evidence is absent in the image. (Perturbations magnified).

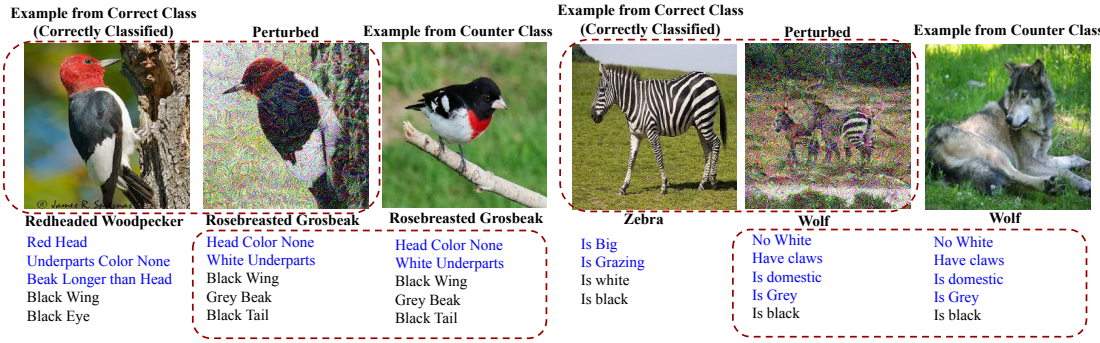


Figure 13: Untargeted: Qualitative analysis of change in attributes due to directed perturbations with a standard network. The attributes are ranked by importance for classification. Most discriminative attributes for clean images correspond to the ground truth class while, those for the perturbed image they compatible with the counter class thus explaining the misclassification. (Perturbations magnified for better visibility).

visually dissimilar counterclasses. Hence, it becomes difficult to ground predicted attributes on perturbed images because there is no visual evidence present for those attributes in the images. For instance in figure 14 first example, "White Throat", "Bill length same as head" and "Solid Back" were responsible for misclassification into the "White breasted kingfisher" class, but as there is no visual evidence available for these attributes in the image originally belonging to "Black billed Cuckoo" class therefore, none of the attributes could be grounded on the perturbed image. Hence, our results show that the visual explanations provided by untargeted perturbations are much more useful for human understanding as compared to targeted perturbations.

2.4.4 *Attribute-based Explanations: Robust Network*

We perform the same experiments with a robust network to study the change in attribute values such that the class of the perturbed image changes back to the ground truth class.

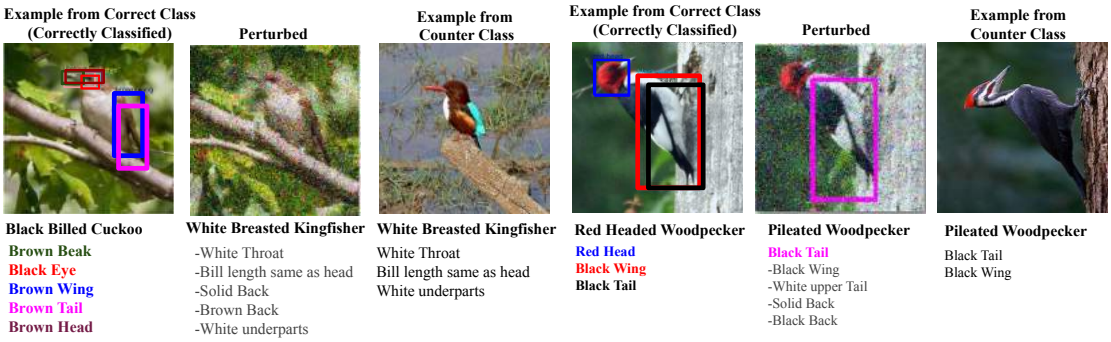


Figure 14: Targeted: Qualitative analysis of change in attributes due to directed perturbations with a standard network. The attributes are ranked by importance for classification. Grounded attributes are color coded for the visibility. Those in gray color could not be grounded. Those attributes common among ground truth and counter class are grounded while those for which no visual evidence is found in the image could not be grounded on the perturbed image, hence, indicating the change in the class. (Perturbations magnified for better visibility.)

By Performing Classification based on Attributes

Untargeted Attacks. Our evaluation on the standard and adversarially robust networks shows that the classification accuracy improves for the adversarial images when adversarial training is used to robustify the network, Figure 9 (purple curves). For example, in Figure 9 for AWA the accuracy of the general classifier improved from 70% to 92% and for LAD it improved from 50% to 78% for adversarial attack with $\epsilon = 0.01$. As expected for the fine-grained CUB-dataset, the improvement is $\approx 31\%$ higher than the AWA and LAD datasets. However, for the attribute-based classifier, the improvement in accuracy for AWA ($\approx 18\%$) is almost double and for LAD ($\approx 22\%$) almost triple that of the CUB-dataset ($\approx 7\%$). This demonstrates that, attributes retain their discriminative power for explanations with the standard as well as robust networks.

Targeted Attacks. Results for the performance of standard and adversarially robust networks against targeted attacks show that the performance of the network improves for adversarial images when tested on an adversarially robust network, Figure 10 (purple curves). Different from untargeted attacks for targeted attacks, the improvement in the performance is not significant. For example, in Figure 10 at $\epsilon = 0.06$ for AWA dataset the accuracy improved to $\approx 12\%$, for CUB it improved to $\approx 14\%$ and for LAD dataset it improved to $\approx 16\%$ while with untargeted attacks the improvement in the accuracy at $\epsilon = 0.06$ is more than double of that with targeted attacks. This shows that when images are misclassified into visually dissimilar classes, it becomes difficult to correctly classify them with robustification as compared to images misclassified into visually similar classes.

Similarly, for attribute-based classifier the improvement in the accuracy is less for targeted attacks as compared to the untargeted attacks, Figure 10 (purple dotted curves). The overall behavior in the improvement of performance for each dataset with targeted attacks is similar to that of untargeted attacks. For instance, at, $\epsilon = 0.06$ the improvement in the accuracy for the CUB-dataset is the least $\approx 11\%$ following AWA $\approx 16\%$ and LAD $\approx 21\%$ datasets. This supports our argument that in order to change the class of

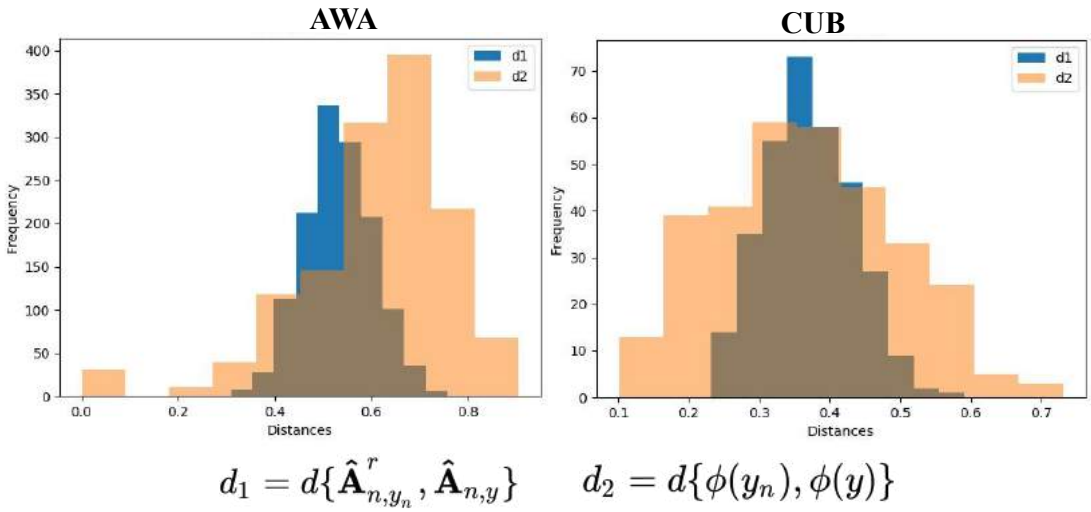


Figure 15: Attribute value distance plots for only adversarial images with and without a robust network. The similarity with the plots in the Figure 11 shows that adversarial image attributes in the presence of a robust network indicate to the ground truth class.

fine-grained images, more number of attributes need to be changed. Overall, our results reveal that even for an adversarially robustified network, untargeted attacks provide better explanations as compared to targeted attacks.

By Computing Distances in the Embedding Space

We also compare the euclidean distance between predicted attributes for only adversarial images in the presence of a standard network and a robust network, as shown in Figure 15. The results reveal that with only adversarial images on robust and standard networks, we observe the same distance distribution as in Figure 11. Thus, attributes explain the correct classification of adversarial images in the presence of the robust network.

Qualitative Analysis

Finally, our analysis with correctly classified images by the adversarially robust network shows that, adversarial images and their predicted attributes with the robust network behave like clean images and their predicted attributes as shown in Figure 16. This also demonstrates that the attributes for adversarial images classified correctly with the robust network still retain their discriminative power and provide complementary explanations.

2.4.5 *Example-based Explanations*

In the final experiment, we demonstrate our visual example and counterexample-based explanations when the attribute values change with directed perturbations. For instance in Figure 17 when an image is classified correctly, besides explaining the classification decision with attributes we enhance our explanations with the complemental example retrieved based on these attributes. Similarly, when an image is misclassified into a counter class, we also enhance our attribute-based explanations by retrieving an image from the counter class.

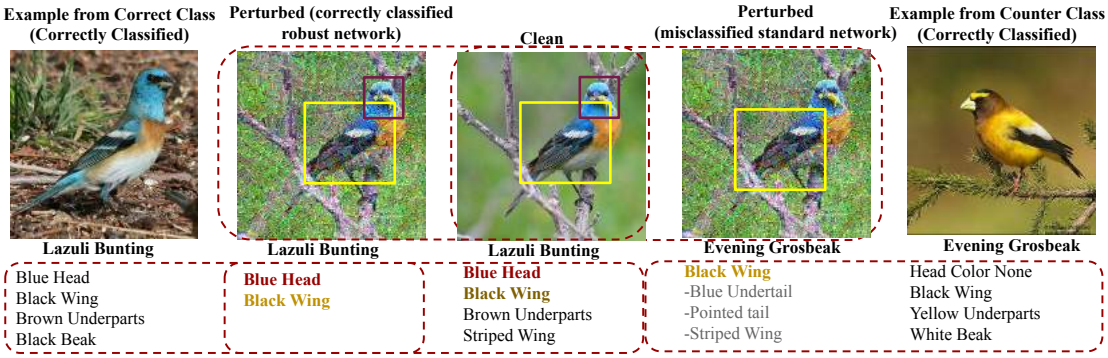


Figure 16: Qualitative analysis for change in attributes due to directed perturbations with a robust network. The attributes are ranked by importance for the classification decision, the grounded attributes are color coded for visibility (the ones in gray could not be grounded). The overlap between the attributes of an adversarial image with a robust network and a clean image with a standard network shows that with a robust network, attributes change back to the ground truth class. (Perturbations magnified for better visibility).

Figure 17: Qualitative analysis for Example-based explanations. Note that when "green eyes, needle shaped bill" changes to "olive eyes, dagger bill" the class of the image changes. These attributes are also complemented with the image-based examples retrieved with these attributes. (Perturbations magnified for better visibility).

Figure 18 reveals the importance of counterexample selection through attributes. In this example both the clean images in first and second row belong to the same class, the "Mallard". However, the clean image in the first row is male Mallard and in the second row is female Mallard, they differ visually. Similarly, the male and female birds of the counterclass "Redbreasted Merganser" also differ visually. The results for the example retrieval for both male and female mallard show that, when images are retrieved through attributes for the male Mallard the retrieved images are male Redbreasted Merganser, while for the female Mallard the retrieved images through attributes are female Redbreasted Merganser. However, when we retrieve the images randomly from the counterclass then the visual similarity can not be ensured. Hence, our attribute-based example selection method selects the visually similar examples to provide the distinction between a clean image and a counter image from the counter class under the presence of intra-class variation.

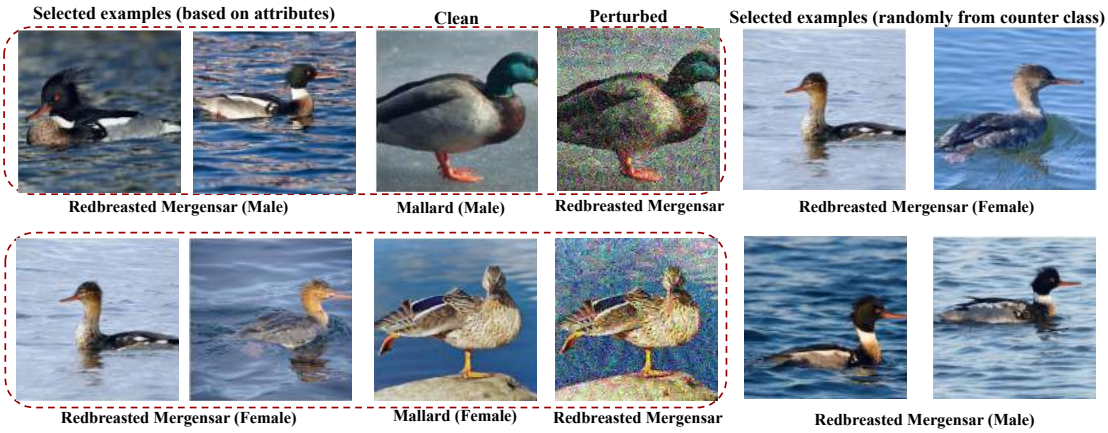


Figure 18: Qualitative analysis for Example-based explanations. Note that both "Mallard" and "Redbreasted Mergensar" classes have intra-class variability, as the male and female birds in both classes look visually different. When we use attributes for retrieving image examples, male Mallard retrieves male Redbrested Mergensar and female Mallard retrieves female Redbrested Merganser, thus incorporating the intra-class variability. (Perturbations magnified for better visibility).

2.5 DISCUSSION AND CONCLUSION

In this work we focused on providing the understanding of neural networks decisions by exploiting counterattributes as well as counterexamples which lead to the misclassification in the counterclass.

Firstly, we showed that attribute-based classifiers perform equally well as direct classifiers. We also showed that the importance of attributes for providing explanations is higher for the fine-grained classification as compared to coarse-grained classification because the distinction between two coarse-grained classes can be made through a single attribute as compared to the fine-grained classes which require numerous attributes for distinction between them.

Secondly, we demonstrated that by introducing adversarial perturbations in the images we were able to change the attribute values to those of counterclass attributes and hence provided counterattribute-based explanations. Our results showed that these attributes contain crucial characteristics for the discrimination between classes.

Thirdly, we repeated all the experiments for the images with perturbations introduced through targeted attacks. Our results showed that, our attribute-based explanations work better with untargeted attacks as compared to the targeted attacks.

We also showed that when a network is robustified against adversarial perturbations, the predicted attribute values for the perturbed images start indicating back towards the correct class, which further confirmed our attribute-based explanations.

Finally, we demonstrated our attribute-based explanations by providing causal reasoning "because the image contains these attributes, therefore it is classified into this class". We also assisted our counterattribute-based explanations with counterexamples selected based on predicted attributes, and showed that our method selected the most precise and illustrative examples even in the presence of intra-class variations.

Hence, we conclude that attributes provide intuitive factual and in the presence of perturbations counterfactual human understandable explanations, especially for fine-grained classification. These explanations could also be enhanced by retrieving visual examples through them. Attributes retain their best disriminative power in the presence of untargeted attacks with standard as well as robustified networks.

3

WIGGLING WEIGHTS TO IMPROVE ROBUSTNESS

3.1 INTRODUCTION

As robustness is essential for building trust in new technology, we consider a technique to robustify networks against unwanted perturbations.

The accuracy of neural networks is heavily affected by Gaussian noise or blur in the image [9], [30], [102]. Occlusion or color saturation will have a similar effect on the network's performance. To achieve robustness, some train the network with perturbed images rather than with clean images [109], or with images perturbed by a learned noise generator [106]. Others [105] propose to train the network with images from a generative model, while [143] show that training with noised perturbations helps against high-frequency. In the same category of approaches, it remains an undecided question whether adversarial training [51] is capable of providing robustness to a broad class of natural perturbations [147], [38], [55].

This work aims to provide an alternative to all these data-side approaches to implement robustness. Instead, we modify a given network by considering transformations to the image filters. We wiggle the weights, see Figure 20, to implement robustness. Compared to data augmentation, apart from delivering better results on general robustness, we also have the advantage of providing built-in robustness, where there is no need to change the data. And, we demonstrate that data augmentation is complementary to our approach, providing a small further improvement in robustness.

For the actual transformations of weights, we are inspired by [71], [42], [26], [25]. Their goal is to build-in transformations to achieve geometrical equivariance. These transformations are robust against global perturbations like rotation and scaling. For our

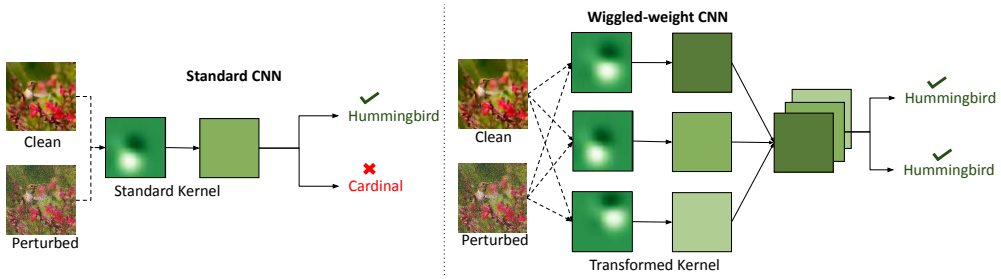


Figure 19: A standard neural network (left) and our wiggled-weight network (right) at test time for classification. The wiggled-weight network has integrated perturbation transforms, which allow for better robustness against perturbations in the input without specialized training.

purpose, we apply multiple stochastic versions of an elastic transformation to make the network more robust in general. We evaluate the resulting wiggled-weight networks on both seen perturbations (rotation-scaling and elastic), as well as unseen perturbations (occlusion, snow, Gaussian noise, Gaussian blur) as presented in [63]. Of the list in the reference, we select six as a good sample of realistic, real-world perturbations, see Figure 23.

Following [106], [63], we select the Resnet architecture [58] as the model to test our approach. To permit a fair comparison, we tune all perturbations such that the decay in the original classification performance is equal for all.

- We propose wiggled-weight convolutions to integrate local perturbations in networks for the purpose of enhancing their general robustness.
- We demonstrate a substantial general robustness of our method on perturbations *seen* during training and, more importantly, also on perturbations *unseen* during training. This includes both natural and adversarial perturbations.
- The general, unseen, robustness is demonstrated to be significantly better than methods based on data augmentation. In fact, the new method can be improved a little further by combining it with data augmentation. And, we improve the classification performance on clean images, leading to the state-of-the-art performance on the STL-10 dataset, 95.45%, and CIFAR-10, 94.97%.

3.2 RELATED WORK

Vulnerability of Classifiers to Natural Perturbations. In [40], [74] the authors show that neural networks are not robust to translations and rotations. [48] deduce that the performance of neural networks drops significantly as compared to humans when the signal-to-noise ratio of images increases. [31] also concluded that, although neural networks are on par in performance with humans, they fail to perform well in the presence of Gaussian noise or blur, which humans easily handle. Therefore, it is crucial to build robusstness against such perturbations into the classification without degrading the performance of clean images, especially in applications like autonomous driving and health.

Benchmarking Natural Perturbations. To promote the study of robustness against naturally occurring perturbations, a few benchmarks have been proposed [63], [61], [48]. In [63], the authors have introduced an impressively large benchmark for natural perturbations. As some of these may be correlated [86], we select six natural perturbations covering the breadth of styles, see Figure 23. In the reference, the authors have defined five levels of severity for each type of perturbation. These levels are based on the visual effect but not standardized on their effect on the classification accuracy. In this work, we first quantitatively standardize the comparison among different perturbations.

Table 1 shows the significance of our standardization method for fair comparison of robustness. When using the mean square error (MSE) between clean and perturbed images for standardization of perturbations, we see that the MSE shows a large variation in classification performance among different types of perturbations. Especially, the MSE for adversarial perturbations tends to be very small, where natural perturbations

Perturbation	Standardized drop in classification accuracy	MSE
Adversarial	10.22	0.02
Elastic	10.60	54.31
Occlusion	10.24	199.73
Gaussian Noise	10.10	11.79
Wave	10.18	602.61
Saturation	10.40	269.71
Blur	10.51	18.20

Table 1: Significance of standardization on CIFAR-10. To permit fair comparison, perturbation's parameters are tuned in such a way as to standardize the drop in classification performance to approximately 10%. *Standardizing on the basis of the mean square error (MSE) between clean and perturbed images gives a large difference in classification performance, and hence is considered not adequate for fairly assessing robustness.*

tend to need a large deviation to show a similar drop in classification accuracy. This is because adversarial perturbations are generated to misclassify an image while keeping the optical difference between clean and adversarial images to a minimum.

The standardization is done by tuning the inflicted perturbation such that the drop of the accuracy of the network is the same regardless of the type of perturbation, Table 1. This enables a fair comparison among different perturbations and the robustness of classifiers.

Robustness to Natural Perturbations. To improve the robustness against natural perturbations, [109] propose to use batch normalization performed on perturbed images instead of clean ones. Similarly, [123] introduces two different normalization techniques, Selfnorm and Crossnorm, to enhance the robustness against perturbations. [11] also utilized perturbed samples and proposed to rectify batch normalization statistics for enhancing the robustness of neural networks against perturbations. Simultaneously, [107] introduces a noise generator that learns uncorrelated noise distributions, demonstrating that training on noisy images enhances the performance against natural perturbations. [55] trained on images with natural perturbations like occlusions or elastic deformations, while achieving good generalization for many of the unseen perturbations, including adversarial ones. [105] and [136] argue that it is impossible to capture all possible natural perturbations mathematically. Therefore, they use generative models to generate images with perturbations to train the network. Instead of training with perturbed inputs, in this work, we integrate local elastic perturbations into the network as a local approximation to the effect of many perturbations aiming for better general robustness.

Robustness to Adversarial Perturbations. In [122], the authors show that by adding small amounts of crafted noise, *adversarial perturbations*, to images, it is possible to change the prediction of the classifier. Since then, many different forms of adversarial perturbations have been studied [82], [98], [121], [19], [97], as well as the robustification against them [51], [82], [51], [19], [32]. When evaluating our model against adversarial perturbations, we select the strong, undefended attack, the basic iterative attack [82] for generating the adversarial perturbations. Experimentally, similar to our work, [106] focused on robustification against adversarial and natural perturbations by tuning Gaussian and Speckle noise. Instead of generating tuned noise and then training the network, we

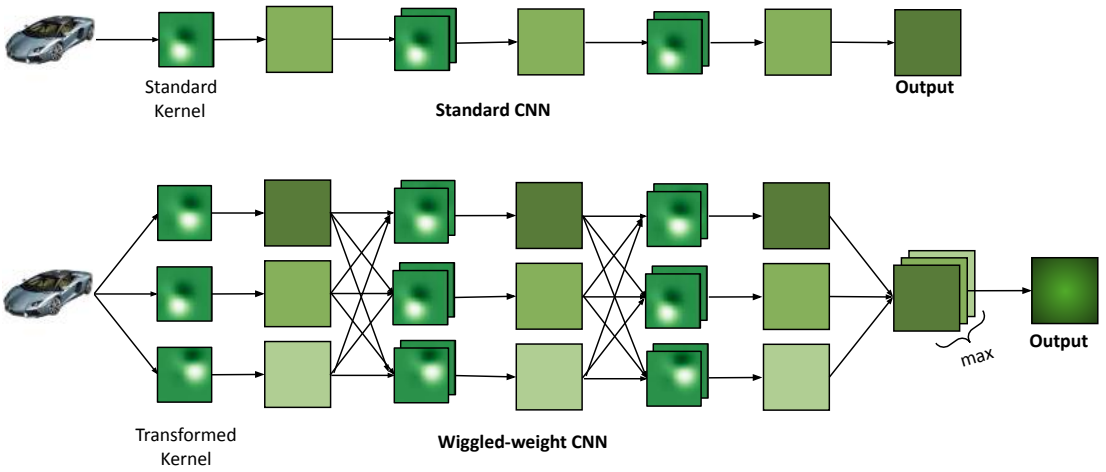


Figure 20: Wiggled-weight Architecture. Top: A standard CNN with three convolutional layers. Bottom: Its wiggled-weight (WWConv) variant. By multiplying the fixed basis with the trainable weights, a single network is transformed into a network with multiple paths, each path with a slightly different basis. At the end, the maximum is selected. This is aimed to provide more robustness against local variations in the input of any kind.

first built-in robustness in the weights of the network and then evaluate on both natural and adversarial perturbations.

Built-in Image Transforms. One of the first methods, suggesting transformations as small units in the network that locally transform their inputs for estimating geometric changes, is in the capsule network architecture [66]. In contrast, in [114] the network is not modified, but rather the update rule of the gradient descent is adapted to learn transformation-invariant weights. Later, both directions evolved [26], [138], [25], [134], [139], [119], [117], [17], where neural networks are equipped with a rotation or scale transformation when they are essential to the task. In [71], [84], [73] neural network modifications are proposed to make them invariant under input transformations.

While these methods consider specific geometric transformations, we focus on local, stochastic elastic perturbations. We demonstrate how they can be incorporated into a CNN for improved robustness.

3.3 METHOD

3.3.1 *Image Transformations*

An image f can be reshaped as a vector $\mathbf{f}$. A wide range of image transformations can be parametrized by a linear operator: scaling, in-plane rotations, shearing. Other transformations, such as out-of-plane rotations, can not be parametrized in an image-agnostic way. However, for small deviation from the original image Taylor expansions

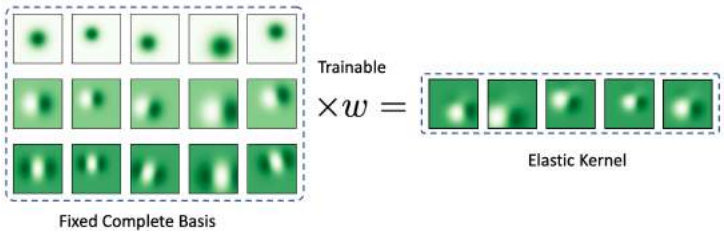


Figure 21: An illustration of how a set of transformed kernels is represented as a trainable linear combination of wiggled-weight fixed basis functions.

can be used, which gives a linear approximation for many image transformations of practical use. Indeed,

$$\begin{aligned} T[f](\epsilon) &\approx T[f](0) + \epsilon \left(\frac{\partial T[f]}{\partial \epsilon} \right) \Big|_{\epsilon=0} \\ &= \mathbf{f} + \epsilon \mathbf{L}_T \times \mathbf{f} = (\mathbf{I} + \epsilon \mathbf{L}_T) \times \mathbf{f} = \mathbf{T} \times \mathbf{f} \end{aligned} \tag{3.1}$$

where T is a transformation, ϵ is the parameter of the transformation and $\mathbf{T}$ is a linear approximation of T for small values of the parameter. For scaling the parameter is the logarithm of the scaling factor, for rotations it is the angle, and so on. $\mathbf{L}_T$ is a matrix representation of an infinitesimal generator of T. An image f can also be viewed as a real-value function of its coordinates $f : x \to f(x)$. We focus here on transformations which can be represented by a smooth field of displacements τ in the space of coordinates. Equation 3.1 can then be rewritten as follows:

$$T[f(x)](\epsilon) \approx f(x + \epsilon\tau(x)) \tag{3.2}$$

We will refer to such transformations as elastic transformations. We will consider them as a linear approximation of a wide range of complex (camera) transformations. All other perturbations can be derived similarly, up to an additive noise.

3.3.2 *Wiggled-weight Convolutions*

Let us consider a convolutional layer Φ parameterized by a filter κ. It takes input image f. The output is:

$$\Phi(f, \kappa) = f \star \kappa = \mathbf{K} \times \mathbf{f} \tag{3.3}$$

where $\mathbf{K}$ is a matrix representation of the filter.

$$\begin{aligned} \Phi(T[f], \kappa) &= T[f] \star \kappa \\ &= \mathbf{K} \times (\mathbf{T} \times \mathbf{f}) \\ &= (\mathbf{K} \times \mathbf{T}) \times \mathbf{f} = \Phi(f, T'[\kappa]) \end{aligned} \tag{3.4}$$

In the most general case, $\mathbf{KT}$ is a matrix representation of a zero padding, followed by a convolution with a kernel and a cropping afterwards. The size of the kernel $T'[\kappa]$ depends on the nature of the transformation T. If the transformation is global, the kernel

can be of a size bigger than the input image. We will consider only the cases when $T'[\kappa]$ is of the same or of a slightly bigger size than the original one.

We propose *Wiggled-weight convolutions*, shortly WWConv, as follows:

$$\text{WWConv} = \max \begin{bmatrix} \beta_0 \Phi(f, \kappa) \\ \beta_1 \Phi(f, T_1[\kappa]) \\ \vdots \\ \beta_n \Phi(f, T_n[\kappa]) \end{bmatrix} \tag{3.5}$$

where β_i are trainable coefficients. We initialize them such that $\beta_0 = 1$, and the rest are zeros. The maximum is calculated per pixel among different transformations of the kernel. At the beginning of training, the operation is thus identical to the original convolution with the same filter. If it is required during training, the other coefficients will activate the corresponding transformations.

3.3.3 *Transformations of a Complete Basis*

In order to apply transformations to filters, we parametrize each filter as a linear combination of basis functions:

$$\kappa = \sum_i w_i \psi_i \tag{3.6}$$

where ψ_i are functions of a complete fixed basis and w_i are trainable parameters. The approach is illustrated in Figure 21. We follow [70] and choose a basis of 2-dimensional Gaussian derivatives.

The transformations when applied to the basis form a transformed basis. Thus, for every transformation from the set, there is a corresponding transform basis. Weights w_i are shared among all bases. We propose a global transformation (rotations-scaling) and a local transformation (elastic) here, and test them on global rotation, global scaling, local occlusions, local snow, Gaussian noise, and Gaussian blur, visual samples of which are shown in Figure 23.

Let us assume that the center of a filter is a point with coordinates $(0, 0)$. For every function from the basis, we first generate a grid of coordinates (x, y). Then we evaluate the value of the function in the coordinates when projected on the pixel grid.

Global Rotation-scaling. In order to transform the functions, we add a small displacement to the coordinates, which leaves the center untransformed. Given a grid of coordinates (x, y), α the deformation intensity and σ be the scaling factor, we define rotation-scaling (See Figure 22, Row top and Row 2) displacements as follows:

$$x' = x + \alpha(x\cos(\theta) + y\sin(\theta)) \tag{3.7}$$

$$y' = y + \alpha(-x\sin(\theta) + y\cos(\theta)) \tag{3.8}$$

where x', y' are the displaced coordinates. And θ is the scale-rotation parameter. When $\cos(\theta)$ is equal to 0 the whole transformation parametrizes rotation. When $\sin(\theta)$ is equal to 0 then it performs scaling. For all other cases, the transformation is a combination of both. The elasticity coefficient controls the severity of the transformations. Thus, for the

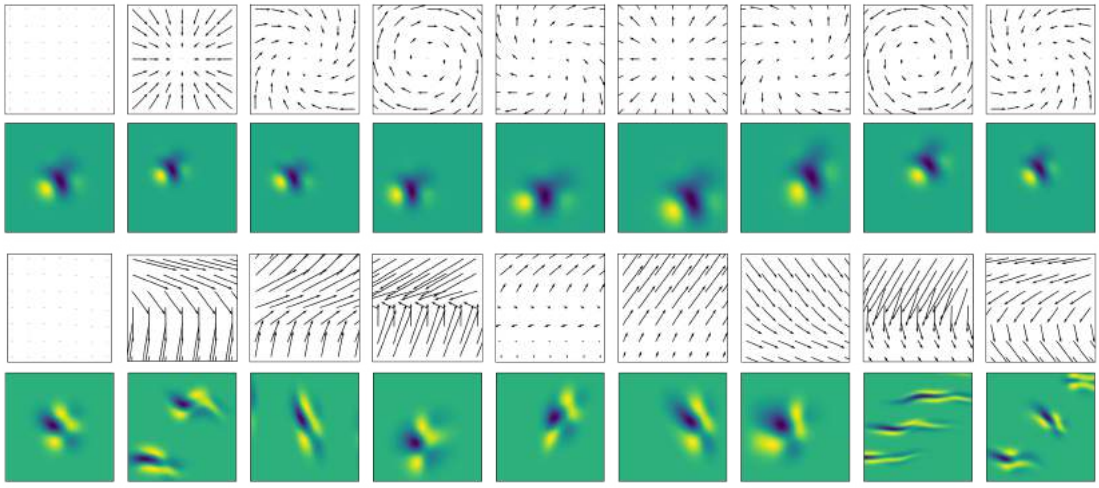


Figure 22: Row top: smooth perturbations for the ***global rotation-scaling transforms****. Row 2: an original filter and its transformed versions. Row 3: smooth perturbations for the* ***local elastic transforms****. Bottom row: an original filter and its transformed versions.*

case of rotation, it is a linear approximation of the sin of the rotation angle. For the case of scaling, α is the scaling coefficient.

Local Elastic Transform. Given a grid of coordinates (x, y), α the elasticity coefficient and σ be the scaling factor we define the elastically transformed filter as following (See Figure 22, Row 3, and Row bottom), i) we take a 2D-affine transform A_θ and map the coordinates (x, y) to the target coordinates (x^t, y^t):

$$A_\theta \begin{pmatrix} x^t \\ y^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{23} & \theta_{33} \end{bmatrix} \begin{pmatrix} x^t \\ y^t \\ 1 \end{pmatrix} \tag{3.9}$$

In order to find θ parameters we select three points in the input grid (x, y) and map them to the output $(x^t = x + U(-\alpha, \alpha), y^t = y + U(-\alpha, \alpha))$. Where, U is the uniform distribution. ii) We get another set of displaced coordinates (x', y') by mapping the coordinates of the kernel as follows:

$$x' = x + \alpha\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}\right) \tag{3.10}$$

$$y' = y + \alpha\left(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}\right) \tag{3.11}$$

iii) Finally, we map the target coordinates (x^t, y^t) to (x', y') using bilinear interpolation.

We follow [119] and use a basis of 2 dimensional Hermite polynomials with the Gaussian envelope for all transforms:

$$\psi_\sigma(x', y') = A \frac{1}{\sigma^2} H_n\left(\frac{x'}{\sigma}\right) H_m\left(\frac{y'}{\sigma}\right) \exp\left[-\frac{x'^2 + y'^2}{2\sigma^2}\right] \tag{3.12}$$

where, A is the normalization constant, H_n is the Hermite polynomial of $n-$th order and σ is the scaling factor. We iterate over n, m-pairs to generate functions.

Figure 23: Sample image from STL-10 dataset showing clean and six different natural perturbations used in our experiments.

3.3.4 *Wiggled-weight Residual blocks*

In order to transform residual networks, we propose a straightforward generalization of the proposed convolution. The standard residual block can be formulated as follows:

$$\text{ResBlock} = f + G(f, \kappa_1, \kappa_2, \dots) \tag{3.13}$$

The block is formulated as follows:

$$\text{WWResBlock} = f + \max \begin{bmatrix} \beta_0 G(f, \kappa_1, \kappa_2, \dots) \\ \beta_1 G(f, T_1[\kappa_1], T_1[\kappa_2], \dots) \\ \vdots \\ \beta_n G(f, T_n[\kappa_1], T_n[\kappa_2], \dots) \end{bmatrix} \tag{3.14}$$

Transformed kernels in the network architecture are shown in the Figure 20.

3.3.5 *Weights Transfer*

To train neural networks successfully, initializing neural networks with Imagenet pre-trained model weights is common. In our case, it is not straight forward to transfer the weights of a standard network to our WWConv network because the network is composed of fixed bases and trainable weights, i.e. multiple parallel networks connected to one another, see Figure 20. Inspired by [118] we assume that in WWConv there is a subnetwork which is identical to the standard network, permitting the transfer of weights from the standard to the WWConv subnetwork. We initialize all weights responsible for inter correlations to zero. Now, the WWConv network until the WWConv max pooling layer (equation 3.14) consists of several parallel networks disconnected to one another. As the filter sizes of the convolutional layers of WWConv match with the sizes of the standard network, we can initialize them with the corresponding Imagenet weights in the standard network. 1×1 convolutions of the standard network and the WWConv network are identical, and therefore, we copy the weights from the standard network to the WWConv one.

3.4 EXPERIMENTS AND RESULTS

Data. Two datasets, CIFAR-10 32×32 pixels and STL-10 96×96 pixels are used in our experiments. CIFAR-10 consists of ten classes with 50000 training and 10000 test images [80]. STL-10 contains 5000 training and 8000 test images in ten categories [24].

Test Input \ Network	**Standard**	**Rotation-scaling wiggling**	**Elastic wiggling**
CIFAR-10 Clean	92.53	94.50	**94.97**
STL-10 Clean	84.40	94.12	**95.45**

Table 2: Classification accuracy on clean images. WWConv networks achieve an improvement in the performance on clean data for CIFAR-10, and a significant improvement of $\approx 11\%$ for clean STL-10 data. The new WWConv network contributes even in the clean image accuracy.

Apart from the data augmentation experiment, the only data augmentation while training is a randomized horizontal flip.

Implementation Details. We use Resnet-152 as the baseline network, SGD optimizer with the cyclic learning rate scheduler at a rate of 0.05 is used for training. For both datasets, we experimented by wiggling the weights of multiple blocks in the Resnet architecture with elastic and rotation-scaling convolutions one-at-a-time. When restricting the wiggling to the first block consisting of multiple non-linear layers [58], we found it to be sufficient to achieve good results.

Evaluating the Standard Network We begin training standard networks for each dataset on clean images. We fine tune the Resnet networks pre-trained on Imagenet and achieve 92.53% and 84.40% for CIFAR-10 and STL-10 clean test set respectively, see Table 2.

Standardizing Network Robustness While considering standard networks as the baseline, we standardize the comparison among robustness of different networks by setting the desired drop at 10% for each dataset, shown in Table 3. We succeed in reaching a standardized drop within a maximum standard deviation of 0.44. Hence, our standardization enables fair comparison among robustified networks on different types of perturbations.

3.4.1 *Evaluating Wiggled-weight Convolutional Network*

We train each classifier network with wiggled-weight convolutions. For both datasets, we initialize the weights of WWConv Resnet-152 with Imagenet weights and fine-tune it. We evaluated our method by adding WWConv with four stocahstic versions of the transform, as shown in Figure 20.

Evaluating Robustified Networks on Clean Images

On clean CIFAR-10 test set, elastically transformed convolutions showed the best performance with an improvement of 2.44%, rotation-scaling following it with an improvement of 1.94%, see Table 2.

For STL-10, the improvement in the performance with elastically transformed convolutions is significant, leading to an improvement of 11.05% and rotation-scaling following it with an improvement of 9.72%, see Table 2. We contend that the reason behind the significant improvement in the performance for STL-10 dataset is that STL-10 is a small dataset, and our wiggled convolutions provide variations in the network, which leads to an improvement in the performance especially for small datasets.

Pertubed Input \ Network	**Standard**	**Rotation-scaling wiggling**	**Elastic wiggling**
		CIFAR-10	
Rotation-scaling	82.81	82.73	**86.59**
Elastic	82.61	85.62	**87.12**
Object Occlusion	81.90	78.35	80.18
Gaussian Blur	82.60	87.96	**90.03**
Gaussian Noise	81.47	88.96	**89.38**
Snow Occlusion	82.81	83.86	**85.23**
		STL-10	
Rotation-scaling	73.98	89.48	**92.29**
Elastic	73.88	87.96	**90.94**
Object Occlusion	73.28	74.56	**81.90**
Gaussian Blur	73.86	87.41	**91.01**
Gaussian Noise	73.60	90.70	**92.60**
Snow Occlusion	73.49	**90.49**	85.63

Table 3: Classification accuracy on perturbed images. WWConv for seen rotation-scaling and elastic, and for four unseen naturally perturbed image styles. For a standard network, we drop the performance to a standardized level by tuning the perturbations. WWConv recovers the drop in the performance for all the perturbations on CIFAR-10 except Occlusion. For STL-10, we recover the classification accuracy on all perturbations. Hence, WWConv significantly enhances the robustness against natural perturbations. Bold shows row-wise best.

Evaluating Robustified Networks on Seen Naturally Perturbed Images.

Table 3 compares the performance of the standard versus the wiggled-weight networks on naturally perturbed images. It uses the same deformation model in the WWConv network as the perturbation applied to the input images, of which sample test perturbations are shown in Figure 23.

We observe on the first two rows in the Table 3 under rotation-scaling and elastic perturbations, that for CIFAR-10, our elastically wiggled convolutions recover the drop on both elastic and rotation-scaling perturbations. On the other hand, on STL-10, our modified convolutions recover the drop for the two seen perturbations, with elastically transformed convolutions showing the best performance when tested on elastic perturbations. Hence, our proposed wiggled-weight convolutions are generally robust against natural perturbations based on the same transformation in the network as the one applied to the image, "seen" perturbations.

Evaluating Robustified Networks on Unseen Naturally Perturbed Images.

We consider the occlusion, Gaussian blur, Gaussian noise, and snow perturbations not explicitly covered by an elastic transformation, hence "unseen during training", for evaluating our new model.

On the CIFAR-10, we observe that both wiggled-weight networks recover the drop in the performance for all the perturbed unseen inputs except occlusions, see Table 3. Elastic transform convolutions show a better recovery in the drop on unseen perturbations,

Perturbed Input \ Network	Standard	Rotation-scaling wiggling	Elastic wiggling
		CIFAR-10	
Adversarial (standardized)	82.13	86.42	**86.49**
Adversarial ($\epsilon = 2$)	28.48	28.27	28.48
		STL-10	
Adversarial (standardized)	71.91	88.12	**91.72**
Adversarial ($\epsilon = 2$)	27.04	40.35	**50.06**

Table 4: Classification accuracy on adversarially perturbed images. As before, for a standard network, we drop the performance by adding adversarial perturbations to cause $\approx 10\%$ drop and a higher drop with $\epsilon = 2$. For CIFAR-10, the elastic WWConv recovers the drop in performance. For STL-10, also elastic WWConv recovers best at $\approx 23\%$. WWConv enhances robustness against adversarial perturbations.

Perturbed Input \ Network	Standard	Elastic Wiggling	Elastic augmentation	Elastic Wiggling+ Elastic augmentation
Clean	92.53	**94.97**	90.54	93.65
Rotation-scaling	82.81	86.59	81.61	**88.04**
Elastic	82.61	87.12	90.79	**89.68**
Object Occlusion	81.91	80.18	**83.62**	75.32
Gaussian Blur	82.60	90.03	90.10	**91.61**
Gaussian Noise	81.47	**89.38**	72.51	85.74
Snow Occlusion	82.81	**85.23**	82.51	83.41

Table 5: Comparing classification accuracy with data augmentation. Comparing the standard, our WWConv, data augmented combined with WWConv networks on naturally perturbed CIFAR-10, we observe that combining WWConv with the data augmentation further enhances network robustness for seen perturbations while the accuracy of elastic WWConv alone is the best one for unseen perturbations.

e.g., for Gaussian Blur it recovers the drop of 9.43%, for Gaussian noise 7.91% and for snow occlusions 2.24%. Thus, the proposed wiggled-weight convolutions show robustness against unseen natural perturbations, with the one exception for occlusions for the CIFAR-10 dataset. The lack in the recovery due to occlusions are ascribed to the size of the CIFAR-10 images, making it difficult for the networks to recover the information lost in occlusion.

On the STL-10, in Table 3 we also test our wiggled-weight networks on five different natural unseen perturbations. We observe that both the wiggled-weight networks recover the induced drop in the performance on unseen perturbations. Similar to CIFAR-10, the elastic WWConv network shows a better recovery on unseen occlusions (8.62%), Gaussian blur (17.15%) and Gaussian noise (19.00%) perturbations. The rotation-scaling WWConv network performs best on the unseen snow perturbations, as well as on the seen global transforms. In contrast with CIFAR-10, on STL-10 the proposed model shows significant recovery in the drop for occlusion perturbations. We conclude that wiggled-weight convolutions show a much better general robustness on unseen naturally perturbed images.

Test Input \ Network	**Resnet**	**WWConv Elastic (1-pert)**	**WWConv Elastic (4-pert)**
Clean	92.53	94.87	**94.97**
Rotation-scaling	82.81	82.81	**86.81**
Elastic	82.61	84.86	**87.12**
Object Occlusion	81.91	79.90	80.19
Gaussian Blur	82.60	87.82	**90.03**
Gaussian Noise	81.47	90.01	**90.51**
Snow Occlusion	82.81	84.88	**85.23**

Table 6: Comparing classification accuracy for network variations. Comparing on CIFAR-10, WWConv Elastic (1-pert) with WWConv Elastic (4-pert). Where 1-pert uses one style of elastic filter, while 4-pert uses four different styles of elastic filters in the network. We observe that varying filter transforms enhances the robustness of the network.

Evaluating Robustified Networks on Adversarially Perturbed Images.

In Table 4, we contrast the performance of a standard network with our WWConv for adversarial images. We test the performance for a standard 10% drop and for high drop with high intensity adversarial perturbations, i.e., $\epsilon = 2$.

On the CIFAR-10, we observe that the wiggled-weight networks are capable of counteracting modest adversarial perturbations, causing a standard drop of 10%. While for adversarial perturbations with the $\epsilon = 2$, causing a drop of $\approx 64\%$, both networks do not add very much anymore, attributed to the size of CIFAR-10 images. Hence, we conclude that the proposed robustified networks help against common adversarial perturbations on CIFAR-10.

For STL-10 dataset, our wiggled-weight network with both perturbation types show resistance against adversarial perturbations. On adversarial perturbations which cause a drop of 10% elastic augmented convolutions show the best resistance with an improvement of 19.81%. Similarly, for a drop of $\approx 57\%$ with the $\epsilon = 2$ elastic augmented convolutions show the best recovery of $\approx 23\%$, with rotation scaling following it with a recovery of $\approx 13\%$. Thus, our wiggled-weight networks also defend adversarial perturbations for both low and high drops.

3.4.2 *Ablation Studies*

Combining with Data Augmentation. To evaluate the effectiveness of WWConv when combined with the data augmentation, we train a WWConv elastic Resnet-152 with elastic data augmentation. Table 5 compares WWConv, data augmentation and the two combined. We observe that our WWConv shows the best performance on clean images, WWConv with data augmentation being the second best. For the naturally perturbed test set, WWConv when combined with the data augmentation further enhances the robustness of the network, while data augmentation alone fails to generalize to unseen perturbations.

Network	**Training Time**	**GPU Usage**	**Network Size**
ResNet-152	0.14 s	1.9 Gb	221.9 Mb
WWConv-ResNet-152	0.32 s	9.1 Gb	221.9 Mb

Table 7: Comparing resources on an Nvidia 1080Ti. Our network requires the same disk space as the baseline network and only double the time to train.

Evaluating the Effectiveness of Varying Filter Transforms. To show the effectiveness of integrating a stochastic variety of filter transforms in the network, we train our WWConv with one type of elastic filter transform (denoted as 1-pert) across all the parallel networks and compare it with a network trained using stochastically differing elastic filter transform across each parallel network (denoted as 4-pert) in Figure 20.

In Table 6 we compare the results for WWConv elastic 1-pert with 4-pert. We observe that although both networks consist of an equal number of four filter transforms, variation of the transform helps. The same transform repeated four times gives a worse performance compared to varying transforms.

Comparing Computational Resources.

In Table 7 we compare the computational complexity of our WWConv network with a standard network. While delivering a much better robustness, our network requires no processing of the data for augmentation, the same disk space as the baseline network, only double the time to train, and GPU memory proportional to the number of transforms.

3.5 CONCLUSION

We formulate a method to enhance the robustness of networks for classification against common perturbations such as occlusion, Gaussian noise, Gaussian blur, and snow. The method transforms the network's weights by four different stochastic instantiations of a local elastic transform to cover the local neighborhood by Taylor expansion in the functional space of all classifiers.

To permit a fair comparison in the performance of perturbed images, we first tune the perturbation parameters to the same drop in classification performance.

In this standardized setting, we demonstrate the effectiveness of our method by improving the performance against natural and adversarial perturbations over standard networks. Local elastic convolutions corresponding to viewpoint change deformation generally perform the best. The results show improved network robustness for four common perturbations, not explicitly modeled in the wiggling, hence "unseen" during training. The improvement in robustness is usually by a large margin, even compared to training with data augmentation by the same transform, "seen" perturbations. Our WWConv can be further improved by exposing it to data augmentation.

In the evaluation, we note that our method unexpectedly enhances the network classification accuracy on clean, CIFAR-10, and STL-10 datasets, where a small loss would also have been acceptable.

We conclude that our wiggled weights approach induces good general robustness for the class of such natural perturbations. At the same time, the costs of implementing into the network and the additional computational resources are modest.

4

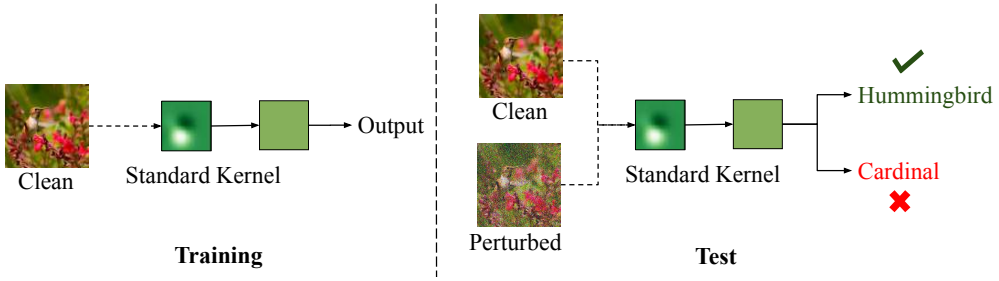
NATURAL PERTURBED TRAINING FOR GENERAL ROBUSTNESS

4.1 INTRODUCTION

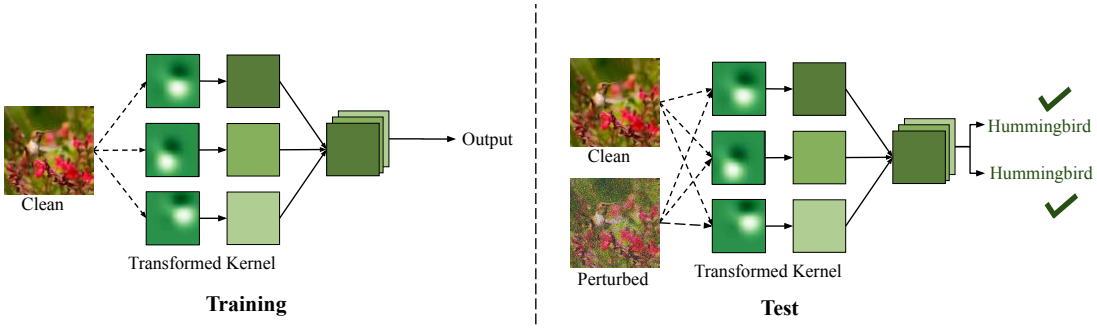
Recent research in machine learning and computer vision shows that changes in the inputs of convolutional neural networks like blur or noise can drastically change the class predictions in the real world [9, 31, 102]. Considering the importance of robustness against natural perturbations, [63] proposed a benchmark consisting of a subset of Image net [27] with corruptions applied to them. Although they introduced five severity levels for each type of perturbation, they do not standardize the effect of the perturbations for a fair, quantitative comparison among the different perturbations. Therefore, in chapter 3, instead of qualitative evaluation, we introduced a standardization procedure to permit a quantitative assessment of robustness among alternative types of perturbations to train a network. We will use this standardization to evaluate our training procedure in this work.

Several methods for the robustness of neural networks against natural perturbations have been proposed in the literature [64,106,143]. [143] hypothesized that Gaussian noise and adversarial training helps against perturbations in the high-frequency domain. [106] showed that by generating properly tuned Gaussian or speckle noise, it is possible to generalize a network to unseen perturbations. To systematically enhance and study the robustness for neural networks against perturbations in this paper, we introduce a simple yet effective training procedure *natural perturbed training*. The network is first trained for n_1 epochs on clean images followed by n_2 epochs on naturally perturbed versions of the same training images. Unlike previous methods, this training method does not require architectural changes, and it is not computationally expensive, while any natural perturbation could be used with it. Moreover, we map filter transforms, proposed in Chapter 3, to input images to explain the effectiveness of integrating natural perturbations in the network, see Figure 35.

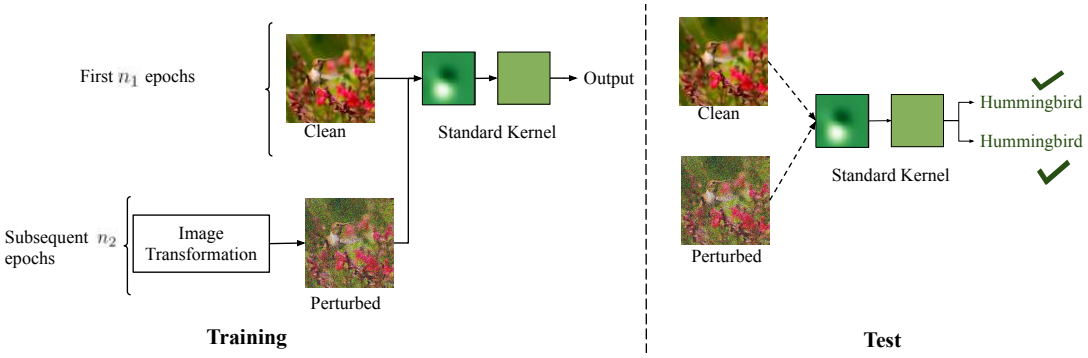
Concurrently, training methods have been introduced to achieve robustness against adversarial perturbations [51, 96, 116]. To date, it is an open problem whether adversarial perturbations help make networks robust against natural perturbations and vice versa [38, 150]. [150] showed that adversarial training helps to reduce the texture bias in neural networks. However, [38] showed that adversarial perturbations do not generalize to natural transformations like translations and rotations. Therefore, in this work, after standardization permits a fair comparison between differently trained networks for robustness, we evaluate whether adversarial perturbations generalize to natural perturbations and the other way around.



(a) Standard CNN.



(b) Wiggled Weight Convolutions.



(c) Natural Perturbed Training.

Figure 24: (24a): A CNN with standard kernels leads to misclassification when tested on perturbed input. (24b): Wiggled weight convolution (WWConv), with transformed kernels, classifies perturbed input correctly. (24c): Similarly, A CNN with a standard kernel but with natural perturbed training classifies perturbed input correctly.

There is also an open debate in literature [120, 126, 149] about the trade-off between robustness and accuracy of clean image classifiers when networks are robustified with adversarial training. We found that our natural perturbed training procedure does not significantly drop the performance on clean images as adversarial training does. For Cifar-10 and STL-10 natural perturbation even helps to improve the accuracy to reach the state of the art performance [119], without the high computational costs of adversarial training.

Standardization is also useful in the evaluation of robustified networks for unseen perturbations. In contrast to [63, 85, 106], we learn the quantitative effect of the type of training for robustness also against *unseen perturbations*.

Our contributions are: (1) We introduce natural perturbed training, which is computationally fast and shows better performance than adversarial training on clean, adversarial as well as natural perturbations. (2) We train neural networks on naturally perturbed images to justify why incorporating natural perturbations in the network enhances robustness, see Figure 35 (3) Natural perturbed training is demonstrated to improve the quantitative robustness of perturbations both seen and unseen during the training. (4) Natural perturbed training even improves the performance of classifiers in the absence of perturbations (without using more data and at almost no costs).

4.2 RELATED WORK

4.2.1 *Natural Perturbations and Robustness*

In [40,74] authors showed that neural networks are not robust to translations and rotations. [48] deduced that the performance of neural networks significantly drops as compared to humans with the increase of the signal-to-noise ratio of images. [31] also concluded that although neural networks are on par in performance with humans, they fail to perform well in the presence of perturbations like Gaussian noise or blur, which humans easily handle. Therefore, it is crucial to building robustness against such perturbations into the classification without degrading the performance of clean images, especially in applications like autonomous driving and health.

To promote the study of robustness against naturally occurring perturbations, a few benchmarks have been proposed [48, 61, 63]. Closely related to our work, in [63] the authors have introduced a large benchmark for natural perturbations, quite a few of which will be correlated [86]. In our work, we selected six more or less independent types of natural perturbations covering the breadth of styles, see Figure 26. In the reference, the authors have defined five levels of severity for each type of perturbation. These levels are based on the visual effect, but not standardized on the classification. As robustness is primarily aimed at the loss of classification performance, in this work, as proposed in chapter 3 at first, we quantitatively standardize the comparison among differently trained networks to analyze the effect on their robustness.

Simultaneously, to improve the robustness against natural perturbations [106] performed data augmentation by carefully tuning Gaussian or speckle noise. [123] introduced two normalization techniques, SelfNorm and CrossNorm, to enhance the generalization for out-of-distribution data. [109] proposed to use batch normalization statistics calcu-

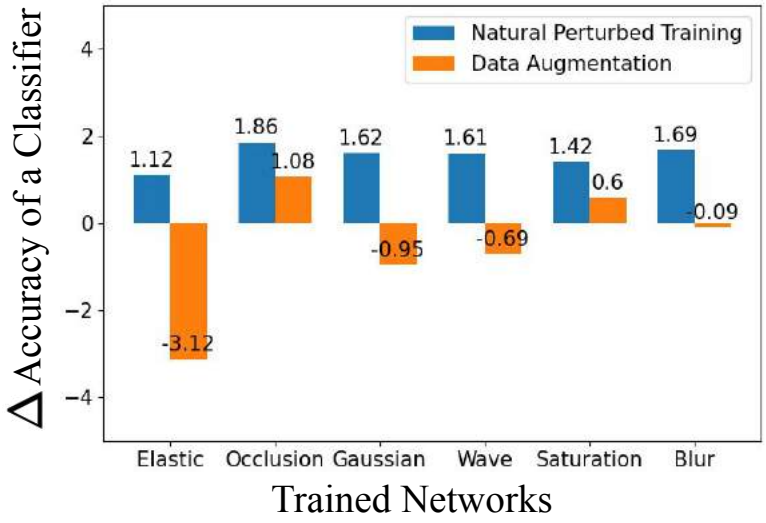


Figure 25: Comparing data augmentation with natural perturbed training on clean images for Cifar-10. △ is the change in accuracy, positive values show improvement, while negative values depict a drop in accuracy. We observe that natural perturbed training shows a better performance on clean images than data augmentation would.

lated on corrupted images instead of clean images to improve the robustness against perturbations. However, all the aforementioned approaches, either require an extra network to find the suitable perturbation or a modification in the network. In contrast, in this paper, we introduce a training procedure in which, after training on clean images, we continue on perturbed versions of the clean inputs and minimize the loss for both of them. This leads to an improvement in the performance on robustness against perturbed images without requiring any architectural changes.

Data augmentation is commonly used to enhance the generalization of deep neural networks. [28] showed an improvement in the generalization by randomly occluding parts of images. [148] trained networks on convex combinations of pairs of images and their labels, which led to an improvement in generalization and robustness against adversarial examples. Similarly, [146] trained on images with regions superimposed from other images. [65] used linear combinations of different data augmentations to enhance generalization.

While these methods enhance the generalization of neural networks, most of these methods train networks on non-realistic images, e.g., superimposing parts of two different images. Furthermore, here we aim to understand the working of the transform augmented convolutional network; we introduce a training procedure using images with perturbed transforms most similar to the built-in transformations of transform augmented convolutional networks as possible.

Note that our natural perturbed training is different from standard data augmentation. Figure 25 contrasts the performance of clean images when the network is trained with the data augmentation versus when it is trained with natural perturbed training. For Cifar-10, we see that natural perturbed training improves clean image classification accuracy for all styles of perturbation. However, data augmentation either leads to a small improvement or even a drop in the performance with elastic, Gaussian, and wave perturbations.

4.2.2 *Adversarial Perturbations and Robustness*

In [122], the authors explored the robustness of neural networks. They showed that by adding small amounts of carefully crafted noise, i.e., *adversarial perturbations* to the images, it is possible to change the prediction of the classifier. Since then, plenty of research [19, 82, 97, 98, 121] has been conducted on finding different types of adversarial perturbations and studying the robustification against them [19, 32, 51, 51, 82]. In this work, we utilize a strong yet undefended attack, i.e., basic iterative method [82] for generating adversarial perturbations. We employ one of the state-of-the-art defense methods, i.e., projected gradient descent [91] in adversarial training, for the comparison with natural perturbed training.

Although adversarial training helps to enhance the performance against adversarial perturbations, [126, 149] showed that with increased robustness of adversarially trained neural networks in classification, simultaneously the network's clean image classification accuracy decreases. This behavior deviates from our natural perturbed training. Apart from increasing the robustness for perturbed image classification, the network retains its accuracy for clean images for most datasets; and even enhances its performance on CUB, StanfordCars, Cifar-10 and STL-10 datasets.

[46] established connections between adversarial and natural perturbation robustness, suggesting that neural networks should be robustified against both of them. [106] focused on robustification against adversarial as well as natural perturbations by using properly tuned Gaussian and Speckle noise. In this work, instead of generating tuned noise and then training the network, we refrain from tuning noise during training. We show that our natural perturbed training offers better performance with occlusion, elastic, and wave than with Gaussian noise as a perturbation.

4.3 METHODS

Given the n^{th} input image x_n and its respective output y_n, a classifier f predicts the class $f(x_n) = y_n$. Here we consider the problem of robust classification against artificially created adversarial ζ^A and natural ζ^t perturbations as noise, motion blur, difference in viewing angle, color saturation, and occlusion.

4.3.1 *Quantitative Standardization*

As the evaluation metric for classification is accuracy, we add perturbations in the input images such that the performance drop ρ in classification accuracy is equal for all perturbations under consideration, as shown in the Table 1. It is given as:

$$\rho = \left[\frac{1}{n}\sum_{n=1}^{N}\mathbb{1}(f(x_n) = y_n)\right] - \left[\frac{1}{n}\sum_{n=1}^{N}\mathbb{1}(f(\zeta^t(x_n)) = y_n)\right] \tag{4.1}$$

where $\mathbb{1}$ is the indicator function. Hence, we set the parameters of each ζ^t under consideration such that the drop ρ is constant for each type of perturbation.

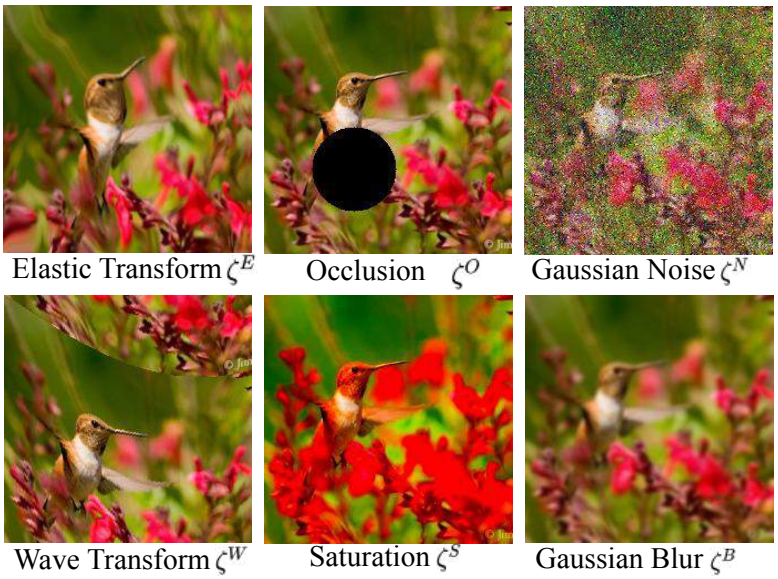


Figure 26: Randomly selected sample images for the six natural perturbations under consideration in our experiments.

4.3.2 *Perturbations*

Natural Perturbations

We consider a set of natural perturbations ζ^t with the least correlations among them, where $t \in \{E, O, N, W, S, B\}$ represents the type of perturbation operator. We create perturbed images by selecting a perturbation from t and applying it on the image $\zeta^t(x_n)$. This leads to a drop in the performance of the classifier $f(\zeta^t(x_n))$. To understand the working of the transform augmented convolutional network; we use perturbed transforms most similar to the built-in transformations of transform augmented convolutional networks as possible. Samples for the six natural perturbations under consideration are shown in Figure 26.

Elastic Transform ζ^E. Elastic deformation usually appears in small variations in the viewing angle of the recording. Given an input image x_n, α the elasticity coefficient and σ be the scaling factor we first generate its coordinates (i, j) and apply the transform as following: i) we take a 2D affine transform A_θ and map the coordinates (i, j) to the target coordinates (i^t, j^t):

$$A_\theta \begin{pmatrix} i^t \\ j^t \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{11} & \theta_{12} & \theta_{13} \\ \theta_{21} & \theta_{22} & \theta_{23} \\ \theta_{31} & \theta_{23} & \theta_{33} \end{bmatrix} \begin{pmatrix} i^t \\ j^t \\ 1 \end{pmatrix} \tag{4.2}$$

In order to find θ parameters we select three points in the input grid (i, j) and map them to the output $(i^t = i + U(-\alpha, \alpha), j^t = j + U(-\alpha, \alpha))$. Where, U is the uniform distribution. ii) We get another set of displaced coordinates (i', j') by mapping the coordinates of the image as follows:

$$i' = i + \alpha(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}) \tag{4.3}$$

$$j' = j + \alpha(\frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{y^2}{2\sigma^2}}) \tag{4.4}$$

iii) Finally, we map the target coordinates (i^t, j^t) to (i', j') using bilinear interpolation, where (i', j') are the displaced coordinates of the image.

Occlusion Transform ζ^O. We apply occlusion transforms by creating a circular mask on the image, M_{c_i,c_j} with the center being c_i and c_j. The values for c_i and c_j are selected randomly from the discrete uniform distribution $U[low, high)$. The radius of the circular mask is a hyperparameter. All the values in the M are set to zero.

Gaussian Noise ζ^N. The Gaussian noise is implemented as follows:

$$(i', j') = (i, j) + \mathcal{N}^\sigma(i, j) \tag{4.5}$$

$$\mathcal{N}^\sigma(i, j) = \frac{1}{2\pi\sigma^2} \exp\left[\frac{-i^2 + j^2}{2\sigma^2}\right] \tag{4.6}$$

Where (i', j') are the perturbed coordinates of the image.

Wave Transform ζ^W. A be the amplitude, B be the frequency and S be the horizontal or vertical shift, we define sinusoidal displacements for a wave transform as follows:

$$i' = \text{Shift}(A\text{Sin}(2\pi S_i B)) \tag{4.7}$$

$$j' = \text{Shift}(A\text{Sin}(2\pi S_j B)) \tag{4.8}$$

Where i', j' are the displaced coordinates.

Saturation Transform ζ^S. Saturation is introduced in the images by converting the RGB image to HSV then increasing the saturation factor, and finally, converting it back to the RGB image.

Gaussian Blur ζ^B. Gaussian blur is introduced by convolving a two-dimensional Gaussian function to the image:

$$(i', j') = (x_n \star \mathcal{N}^\sigma)(i, j) \tag{4.9}$$

$$\mathcal{N}^\sigma(i, j) = \frac{1}{2\pi\sigma^2} \exp\left[\frac{-i^2 + j^2}{2\sigma^2}\right] \tag{4.10}$$

Where $\star$ is the convolution operator. The size of the local neighborhood is determined by the scale (σ) of the Gaussian function.

The natural perturbations are class agnostic in a stochastic sense. However, they are made image specific by selecting different perturbations for different images. For elastic deformation, we vary the intensity of elasticity for each image such that it leads to a specific drop ρ. For occlusion, the position of occlusion is randomly selected for each image, the intensity of Gaussian noise is also randomly uniformly varied. Per image, the wave is scaled uniformly at random, as are the saturation factor and variance of the Gaussian blur filter.

Adversarial Perturbations

Adversarial examples are generated while satisfying two properties, 1) the class of the adversarial image is different from the class predicted for the clean image i.e. $f(\zeta^A(x_n)) \neq f(x_n)$, 2). Perturbed and original images are visually similar, and their similarity is determined by the l_p-norm. While fulfilling these two properties, we use a basic iterative method [82] for generating adversarial examples $\zeta^A(x_n)$. We find the perturbation δ_n with a small norm l_∞ bounded by ϵ such that $f(x_n) \neq f(\zeta^A(x_n))$, where $\zeta^A(x_n) = x_n + \delta_n$ and $\delta_n \leq \epsilon$:

$$\zeta^A(x_n^0) = x_n + \delta \tag{4.11}$$

$$\zeta^A(x_n^{k+1}) = \text{Clip}_\epsilon\{\zeta^A(x_n^k) + \epsilon_s Sign(\nabla_x(\mathcal{L}_r^\delta(\zeta^A(x_n^k), y_n, w))\} \tag{4.12}$$

where $\mathcal{L}_r^\delta(\zeta^A(x_n^k), y_n, w)$ represents the gradient of cost function w.r.t the perturbed image $\zeta^A(x_n^k)$ at step k, ϵ_s determines the step size taken in the direction of sign of the gradient and the result is clipped by ϵ.

4.3.3 *Robustness*

The neural network classifier is trained by minimizing the loss function:

$$\mathcal{L}_s = \min_w \frac{1}{|S|} \sum_{(x_n, y_n) \in S} \mathcal{L}(f(x_n), y_n) \tag{4.13}$$

where $S = \{(x_n, y_n) | x_n \in X, y_n \in Y\}$ is the training set, w the network parameters and $\mathcal{L}$ the cross-entropy loss. Usually, the data augmentation is performed by adding perturbed versions of the input images. The network is trained by replacing the clean input image x_n with its perturbed version $\zeta^t(x_n)$ in Equation 4.13.

Natural Perturbation Robustness

In order to learn better loss surfaces for clean image classification and robustification against perturbed inputs, in this work we introduce *natural perturbed training* as shown in Figure 27. We start training the classifier with clean images x_n for n_1 epochs while optimizing the loss $\mathcal{L}_s$. Then we add their perturbed versions $\zeta^t(x_n)$ besides the clean for the subsequent n_2 epochs while minimizing the loss for both of them, i.e. $\mathcal{L}_r^\zeta = \frac{\mathcal{L}_s + \mathcal{L}^\zeta}{2}$, where $\mathcal{L}^\zeta$ is the loss for perturbed samples. The procedure for natural perturbed training is given in the box 2.

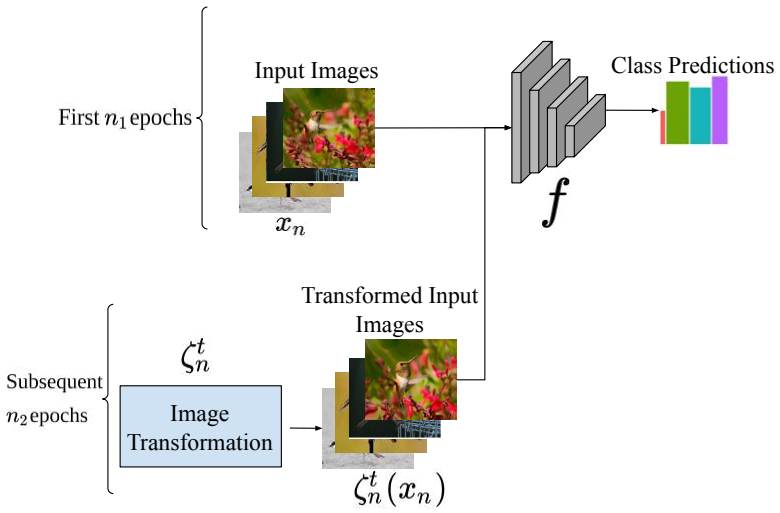


Figure 27: Our natural perturbed training procedure. We train a network f on clean samples x_n for first n_1 epochs. In subsequent n_2 epochs we add perturbed versions of input images $\zeta^t(x_n)$ in the training while optimizing the loss for both clean and perturbed samples for the rest of epochs.

Adversarial Robustness

For adversarial robustness, we consider adversarial training as described in [51]. The network is trained on adversarial samples besides clean images, while the loss function is optimized for both clean and adversarial samples given by:

$$\mathcal{L}^{\delta} = \min_{w} \frac{1}{|S|} \sum_{(\zeta^A(x_n), y_n) \in S} \mathcal{L}(f(\zeta^A(x_n)), y_n) \tag{4.14}$$

$$\mathcal{L}_r^{\delta} = \mathcal{L}_s + \mathcal{L}^{\delta} \tag{4.15}$$

where $\mathcal{L}_s$ is the loss for clean images and $\mathcal{L}^{\delta}$ is the loss for adversarial images.

Algorithm 2 Natural Perturbed Training for Robustification.

Given $S = \{(x_n, y_n) | x_n \in X, y_n \in Y\}$, learning rate η and a set of natural perturbations ζ^t.
Initialize w randomly
for $epoch = 1$ to $n_1 + n_2$ **do**
  for *minibatch* $B \subset |S|$ **do**
    $\mathcal{L}_s = \mathcal{L}(f(x_n), y_n, w)$
    if $epoch > n_1$ **then**
      $\mathcal{L}^{\zeta} = \mathcal{L}(f(\zeta^t(x_n)), y_n, w)$
      $\mathcal{L}_r^{\zeta} = \frac{\mathcal{L}_s + \mathcal{L}^{\zeta}}{2}$
    end if
    Update w with SGD.
    $w = w - \eta \nabla_w \mathcal{L}_r^{\zeta}$
  end for
end for

4.3.4 *Implementation Details*

Evaluation Metric. We use change in the accuracy $\triangle$ as the evaluation metric for the robustness of classifiers. The change is calculated between a standard classifier for clean inputs $f(x_n)$ and a robustified classifier for clean $f_r(x_n)$ or perturbed $f_r(\zeta^t(x_n))$ inputs. The change in the accuracy is given by:

$$\triangle = \left[\frac{1}{n}\sum_{n=1}^{N} \mathbb{1}(f(x_n) = y_n)\right] - \left[\frac{1}{n}\sum_{n=1}^{N} \mathbb{1}(f_r(\zeta^t(x_n)) = y_n)\right] \tag{4.16}$$

where $\mathbb{1}$ is the indicator function.

Standard Network Training and Testing. We perform classification using Resnet-152. For Cifar-10, we train the networks from scratch. For other datasets, networks are pre-trained on Image-net and fine-tuned on the respective datasets. The networks are tested for both clean and perturbed inputs. Natural perturbations are generated using the method described in section 4.3.2 while keeping the drop ρ from equation 4.1 the same for all perturbations to ensure standardization. To make the perturbations diverse across each image, we select the parameters of perturbations randomly. Adversarial perturbations are created using the basic iterative method, with the number of steps K taken as 10 and ϵ values such that the drop ρ is the same as for other perturbations. The metric of similarity between clean and adversarial samples is l_∞ norm.

Robust Network Training and Testing. Networks are robustified with natural perturbed training, see the box 2. Each network is robustified with one type of perturbation and the parameters for perturbations are tuned such that they lead to a constant drop ρ, see Equation 4.1. Adversarial training is performed using projected gradient descent (PGD) with $K = 10$ and ϵ tuned such that it leads to the same drop ρ as the drop of other perturbations. The parameters for the optimizer, learning rate scheduler, and number of epochs are constant across adversarial training and natural perturbed training within a dataset. PGD adversarial training makes $O(KS)$ computational gradient steps in one epoch, where K is the number of steps and S is the dataset size. This procedure is K times slower than the standard training $O(S)$ [137] hence, our perturbed natural training is equally faster than adversarial training.

4.4 EXPERIMENTS AND RESULTS

We compare natural perturbed training with adversarial training on clean, natural perturbed and adversarial inputs. In all plots, a symbol represents one run on a trained network with one specifically (perturbed or clean) test set: the symbol represents the test perturbation type, while the color represents the training perturbation type.

Datasets. Six datasets of varying granularity and size are used in our experiments. Cifar-10 [80] consists of ten coarse-grained classes with 50000 training and 10000 test images. STL-10 [24] contains 5000 training and 8000 test images belonging to ten coarse-grained categories. Different from Cifar-10, the image size is 96×96 pixels. The Large attribute dataset (LAD) [151] contains, 78017 images with 230 fine-grained classes. We use 11702 training, 9947 validation and 9284 test images for our experiments. Animals with attributes (AwA) [140] consists of 37322 images with 50 fine-grained classes. We use

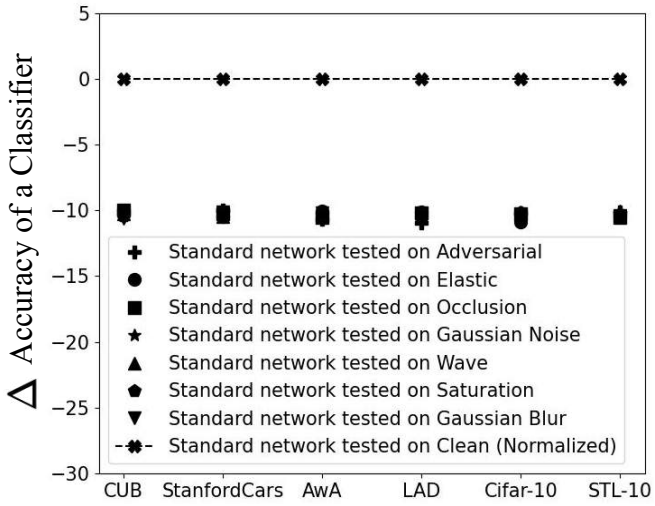


Figure 28: Standardization by calibrating the drop ρ. The cross symbol at zero shows the normalized accuracy of a standard network for clean images. Each of the symbols on -10 shows the standardization by dropping the performance of a standard network when a perturbation is introduced. Hence, the overlap of symbols at -10 for all the perturbations show the degree to which the standardization is uniform.

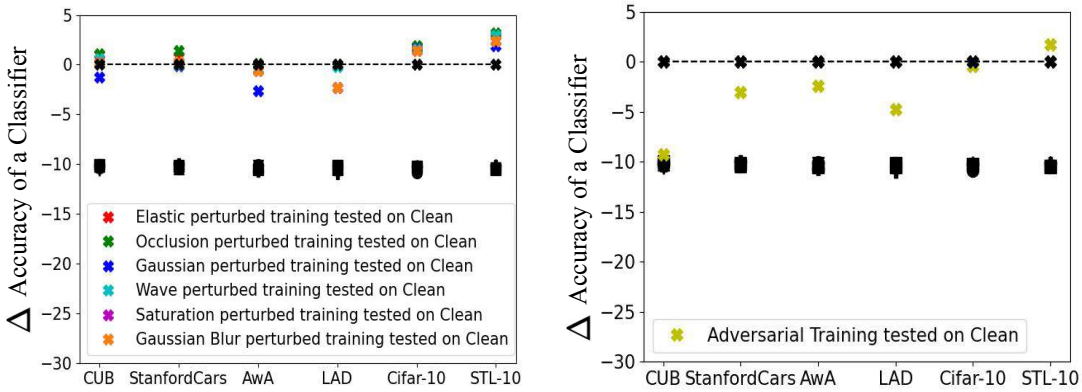
10450 of them for training, 7524 for validation, and 9674 for testing. StanfordCars [78] contains 8144 train and 8041 test images with 196 fine-grained categories of cars. The CUB-birds dataset [135] consists of 11788 images with 5395 for training, 599 for validation and 5794 for testing, divided over 200 fine-grained categories of birds. The input size for all fine-grained datasets is taken as 224×224 pixels.

4.4.1 *Standardizing Network Robustness*

Normalizing Accuracy. We begin by evaluating the performance of a standard neural network classifier for clean images. A standard classifier shows the test accuracy of 93.18 for Cifar-10, 88.60 for STL-10, 87.86 for LAD, 84.79 for AwA, 86.48 for StanfordCars, and 81.20 for CUB dataset. The performance of the standard classifiers for clean images is the reference value of zero, as indicated by the cross symbol, see Figure 28.
Standardization by Calibrating the Drop ρ. While considering the standard networks as the baseline, we standardize the comparison among robustness of different networks by setting the desired drop ρ in Equation 4.1 at 10% for each dataset, shown in Figure 28 at -10%. We succeed in reaching a standardized drop with a maximum deviation of 0.26%. Hence, our standardization enables fair comparison among robustified networks on different types of perturbations.

4.4.2 *Evaluating Robustified Networks on Clean Images*

We contrast the performance of adversarial training with natural perturbed training on the clean test set. Figure 29a shows the performance of a network trained with natural perturbed training and tested on clean inputs. Except for Gaussian blur on LAD and Gaussian noise on AwA and CUB, natural perturbed training retains the performance of



(a) Evaluating Natural perturbed training for clean images.

(b) Evaluating Adversarial training for clean images.

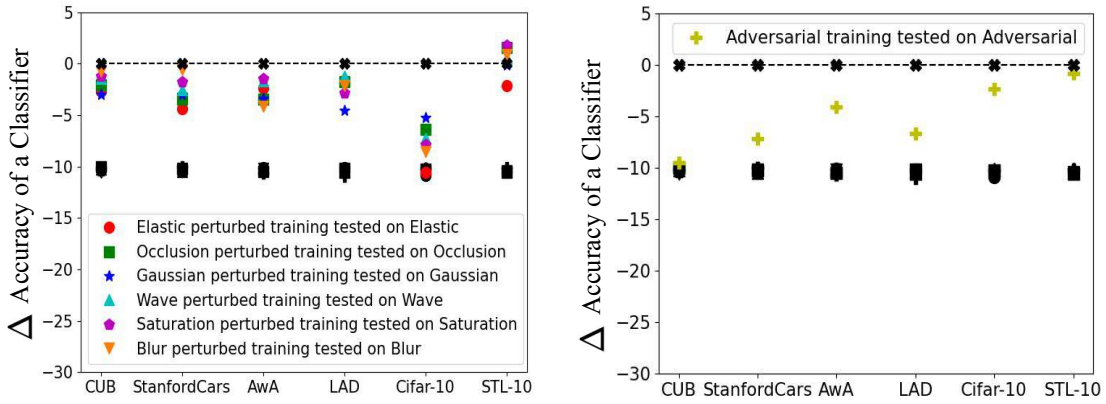
Figure 29: Comparing the performance of natural perturbed training with adversarial training for clean images, where the cross symbol represents a clean test set and the color of the symbol represents the type of training perturbation. Adversarial training degrades the accuracy in the classification of clean images, but natural perturbed training does not degrade the performance on clean images. It even improves the classifier accuracy for four in six datasets.

the classifier on clean images. For CUB, StanfordCars, Cifar-10 and STL-10 datasets, training with the perturbed natural images even leads to an improvement in performance as compared to a standard network trained only on clean images. We achieve a maximum of 95.04 for Cifar-10 and 91.81 for STL-10 with our natural perturbed training. Figure 29b shows the performance of adversarially robustified networks on clean images. We see that robustifying networks against adversarial perturbations leads to the drop in the performance on clean images for all datasets except STL-10. Hence, adversarial training shows a trade-off between robustness on adversarial perturbations and clean image accuracy. In contrast, our natural perturbed training does not degrade clean image accuracy but leads to an improvement in the performance.

4.4.3 *Evaluating Robustified Networks on Seen Perturbations*

We evaluate the robustness of natural perturbed training on the same type of perturbation e.g. a network trained with elastic perturbed training tested on elastic (seen perturbations) as shown in Figure 30a. Results show that natural perturbed training helps to recover the performance when tested on seen perturbations for both coarse and fine-grained datasets. The recovery is highest for STL-10 and least for Cifar-10. Where Cifar-10 and STL-10 are both coarse-grained, the input size in Cifar-10 is around three times smaller than STL-10. Hence we argue that after introducing natural perturbations, the damage in Cifar-10 is too much to recover from. In general, all datasets show significant recovery in the performance with the natural perturbed training in the presence of seen perturbations.

Figure 30b shows the results for adversarial images tested on adversarially robustified networks. We observe that adversarial training helps against adversarial perturbations. However, the recovery in the performance of natural perturbations with the natural perturbed training is higher for all datasets except Cifar-10. Hence, our natural perturbed



(a) Evaluating Natural perturbed training for seen natural perturbations.

(b) Evaluating Adversarial training (AT) for adversarial perturbations.

Figure 30: Comparing the performance of natural perturbed training with adversarial training on seen perturbations. Where the type of the symbol represents the test perturbation type and the color of the symbol represents the type of training perturbation. Adversarial training recovers the performance on adversarial images, but the recovery for natural perturbations with natural perturbed training is higher.

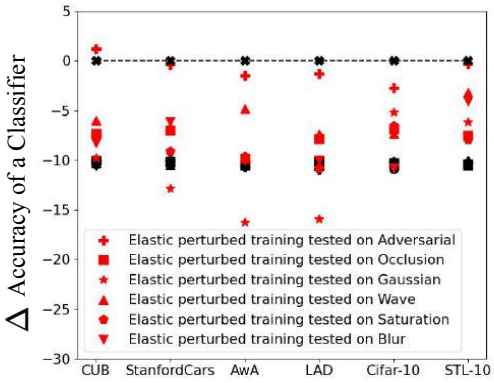
training shows better generalization on perturbation in images seen during training as compared to adversarial training on seen adversarial perturbations.

4.4.4 *General Robustness of Elastic and Occlusion Perturbations*

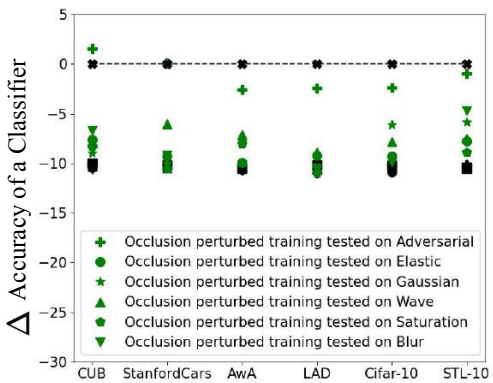
In Figure 31 we contrast the general robustness of natural perturbed training with adversarial training by testing them for unseen perturbations, i.e. perturbations not seen during the training.

Effectiveness of Natural Perturbed Training on Unseen Perturbations. Figure 31a, 31b shows the performance of elastic perturbed training and occlusion perturbed training tested on unseen adversarial and natural perturbations respectively. Results show that robustification with both elastic and occlusion perturbations recover the drop due to adversarial perturbations (plus symbol). We observe that natural perturbed training generalizes to other natural perturbations, except for elastic perturbed training on Gaussian noise for StanfordCars, AwA and LAD (red star symbol). Coarse grained Cifar-10 and STL-10 show the highest recovery on unseen natural perturbations. Hence, our natural perturbed training shows general robustness over adversarial as well as natural perturbations, while being even remarkable for coarse-grained datasets.

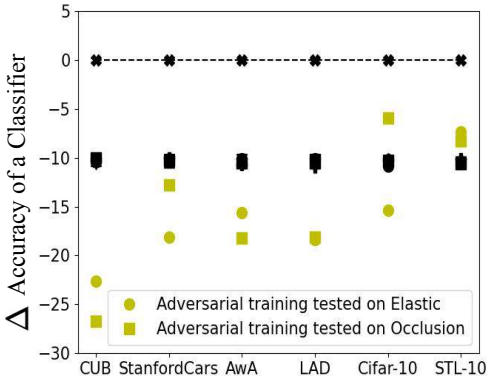
Ineffectiveness of Adversarial Training on Unseen Perturbations. Figure 31c shows the results for an adversarially trained network (depicted by yellow symbols) and tested on unseen natural perturbations elastic (circle symbol) and occlusion (square symbol). Adversarial training does not generalize to unseen natural perturbations for fine grained datasets. It even leads to a further drop in the performance for them. For CUB and LAD, the drop almost doubles. For the coarse grained Cifar-10 dataset it helps against occlusion perturbation and for STL-10 it helps for all perturbations. However, the recovery is smaller than with the natural perturbed training. Hence, natural perturbed training shows better generalization than adversarial training for unseen perturbations.



(a) Evaluating elastic perturbed training for unseen natural perturbations.



(b) Evaluating occlusion perturbed training for unseen natural perturbations.



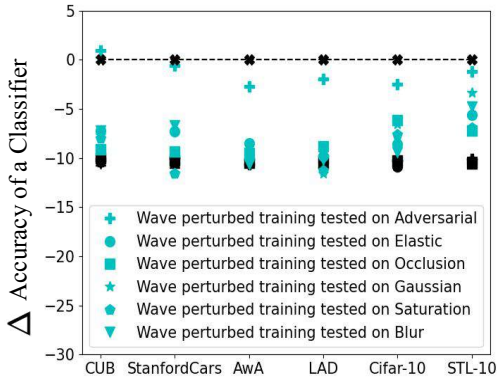
(c) Evaluating Adversarial training for unseen natural perturbations.

Figure 31: Comparing the performance of Natural perturbed training with Adversarial training on unseen perturbations. The type of symbol represents test perturbation and color of the symbol represents the type of training perturbation. Adversarial training shows some general robustness on coarse-grained datasets but for fine-grained datasets it fails to generalize. Natural perturbed training generalizes to adversarial perturbations and other natural perturbations.

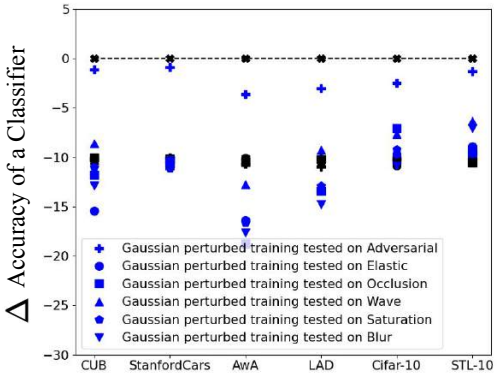
4.4.5 *General Robustness of Wave and Gaussian Perturbations*

In Figure 32 we contrast the general robustness of natural perturbed training with adversarial training by testing them for unseen perturbations i.e. perturbations not seen during the training. Here, we present the results for wave and Gaussian noise.

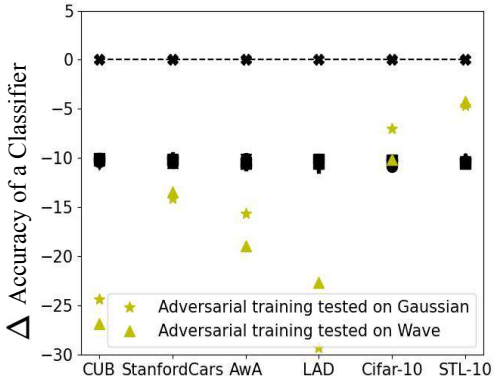
Effectiveness of Natural Perturbed Training on Unseen Perturbations. Figure 32a, 32b shows the performance of wave perturbed training and Gaussian perturbed training tested on unseen adversarial and natural perturbations respectively. Results show that robustification with both wave and Gaussian perturbations recover the drop due to adversarial perturbations (plus symbol). We observe that natural perturbed training especially wave perturbed training generalizes to other natural perturbations too. Gaussian perturbed training also generalizes to other natural perturbations except for CUB, AwA and LAD datasets. Coarse grained Cifar-10 and STL-10 show the highest recovery on unseen natural perturbations. The recovery for coarse grained datasets



(a) Evaluating wave perturbed training for unseen natural perturbations.



(b) Evaluating Gaussian perturbed training for unseen natural perturbations.

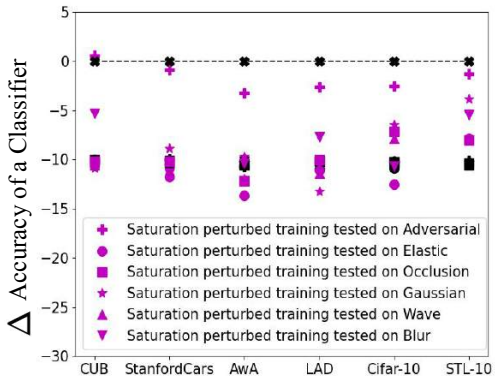


(c) Evaluating Adversarial training for unseen natural perturbations.

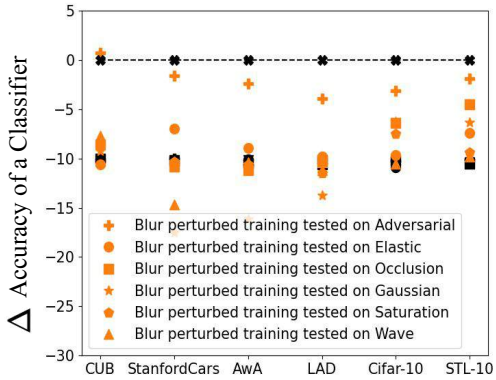
Figure 32: Comparing the performance of Natural perturbed (wave, Gaussian) training with Adversarial training on unseen perturbations. The type of symbol represents test perturbation and color of the symbol represents the type of training perturbation. Adversarial training shows some general robustness on coarse-grained datasets but for fine-grained datasets it fails to generalize. Natural perturbed training generalizes to adversarial perturbations (plus symbol) and other natural perturbations.

with wave perturbations is better than the Gaussian noise. Hence, our natural perturbed training shows general robustness over adversarial as well as natural perturbations, while being even remarkable for coarse-grained datasets.

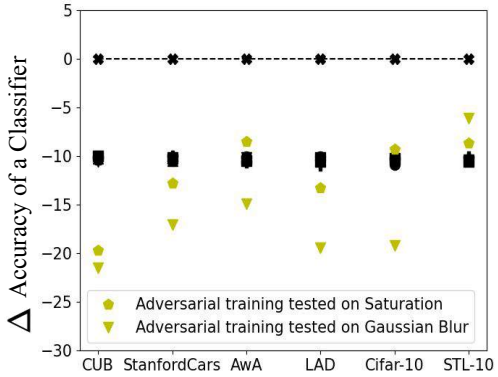
Ineffectiveness of Adversarial Training on Unseen Wave and Gaussian Perturbations. Figure 32c shows the results for an adversarially trained network (depicted by yellow symbols) and tested on unseen natural perturbations wave (triangle symbol) and Gaussian (star symbol). Adversarial training does not generalize to unseen natural perturbations for fine grained datasets. It even leads to a further drop in the performance for them. For CUB and LAD, the drop almost triples. For the coarse grained Cifar-10 dataset it helps against Gaussian perturbation and for STL-10 it helps for both perturbations. However, the recovery is smaller than with the natural perturbed training. Hence, natural perturbed training shows better generalization than adversarial training for unseen perturbations.



(a) Evaluating saturation perturbed training for unseen natural perturbations.



(b) Evaluating Gaussian blur perturbed training for unseen natural perturbations.



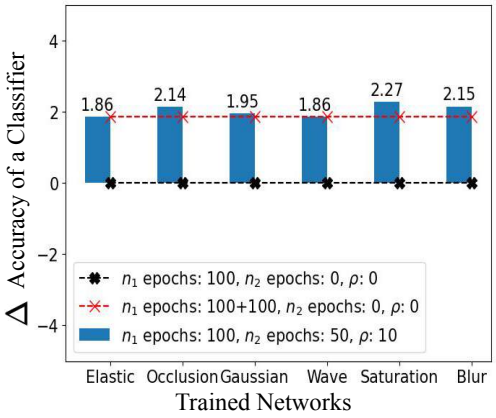
(c) Evaluating Adversarial training for unseen natural perturbations .

Figure 33: Comparing the performance of Natural perturbed (saturation, Gaussian blur) training with Adversarial training on unseen perturbations. The type of symbol represents test perturbation and color of the symbol represents the type of training perturbation. Adversarial training shows some general robustness on coarse-grained datasets but for fine-grained datasets it fails to generalize. Natural perturbed training generalizes to adversarial perturbations (plus symbol) and other natural perturbations.

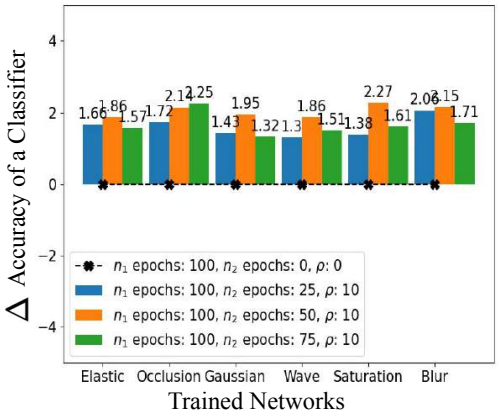
4.4.6 *General Robustness of Saturation and Gaussian Blur Perturbations*

In Figure 33 we compare the general robustness of natural perturbed training with adversarial training by testing them for unseen perturbations i.e. perturbations not seen during the training. Here, we present results for saturation and Gaussian blur.

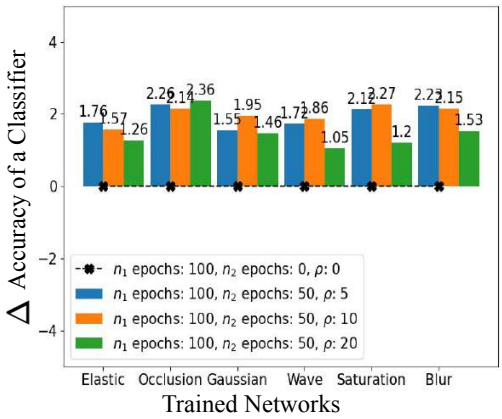
Effectiveness of Natural Perturbed Training on Unseen Perturbations. Figure 33a, 33b shows the performance of saturation perturbed training and Gaussian blur perturbed training tested on unseen adversarial and natural perturbations respectively. Results show that robustification with both saturation and Gaussian blur perturbations recover the drop due to adversarial perturbations (plus symbol). We observe that natural perturbed training generalizes to other natural perturbations too. Except for saturation perturbed on AwA and LAD datasets. Coarse grained Cifar-10 and STL-10 show the highest recovery on unseen natural perturbations. Hence, our natural perturbed training shows general



(a) Standard training for a large number of epochs vs natural perturbed training for less epochs i.e $(n_1 + n_1) > (n_1 + n_2)$.



(b) Evaluation by varying the number of subsequent (n_2) *epochs to 25, 50 and 75.*



(c) Evaluation by varying the drop ρ *to* 5%, 10% *and* 20%.

Figure 34: Ablation on Cifar-10 clean: Figure 34a shows training with perturbed images for a small number of epochs performs better than training with clean for a large number of epochs. Figure 34b shows perturbed training with an average number of epochs performs best. Figure 34c shows a moderate drop of 10% *leads to the best performance on clean images.*

robustness over adversarial as well as natural perturbations, while being even noteworthy for coarse-grained datasets.

Ineffectiveness of Adversarial Training on Unseen wave and Gaussian Perturbations. Figure 33c shows the results for an adversarially trained network (depicted by yellow symbols) and tested on unseen natural perturbations saturation (five pointed star symbol) and Gaussian blur (triangle down symbol). Adversarial training does not generalize to unseen natural perturbations for fine grained datasets. It even leads to a further drop in the performance for them. For CUB, LAD and Cifar-10 Gaussian blur test set the drop almost doubles. For coarse grained Cifar-10 saturation test it neither helps nor degrades the performance. For STL-10 it helps for both perturbations. However, the recovery is smaller than with the natural perturbed training. Hence, natural perturbed training shows better generalization than adversarial training for unseen perturbations.

Input	Clean	Adversarial	Elastic	Occlusion	Gaussian Noise	Wave	Saturation	Blur
△	1.63	1.38	-3.14	-0.62	-1.67	-5.12	-0.17	-3.59

Table 8: Multiple perturbations training. △ shows the change in the accuracy between a standard network and robustified one. Numbers in positive show an improvement in performance, negative show the drop not recovered from the initial 10% *drop. In contrast with the Cifar-10 results in Figure 30a, 31a and 31b we observe a better generalization with multiple perturbations training.*

4.4.7 *Ablation Studies*

We perform ablation studies in Figure 34 on Cifar-10 by varying the parameters of the natural perturbed training. In Figure 34a we compare a standard network trained for 200 epochs on clean images with natural perturbed training by 100 clean and 50 perturbed epochs. Results show that natural perturbed training with a smaller number of epochs achieves better performance for clean images than a standard network with a larger number of epochs.

In Figure 34b we vary the number of perturbed epochs n_2 and test the performance of networks for clean images. Results depict that perturbed training with an average number of 50 epochs performs best. Figure 34c compares the performance of networks trained with different perturbation levels leading to drops of 5%, 10% and 20%. Results show that a moderate drop of 10% leads to the best performance on clean images.

Finally, in Table 8 we evaluate the robustness of a network trained with multiple perturbations applied to the same image during subsequent n_2 epochs. The perturbations are elastic, occlusion, Gaussian noise and saturation. Compared to the results in Figure 30a, 31a and 31b we observe an improvement in the recovery. Hence, training with multiple perturbations helps to enhance the overall robustness of the network.

4.5 CONCLUSIONS

While using, standardization procedure based on the effect of perturbations on the accuracy of the network, a connection between input image perturbations and corresponding transformations of the convolutional filters of a network is established. We introduced a new training procedure for enhancing the robustness of classifiers against perturbations. We provide a rationale behind the modification of the network for enhancing its robustness by training the standard network with similarly transformed images. We demonstrated the effectiveness of our natural perturbed training for clean, adversarial and natural perturbations, both seen and unseen during the training. Our results showed that natural perturbed training, while being computationally fast, also shows better generalization on adversarial and natural perturbations than adversarial training. Moreover, it improves the classifier accuracy on clean images for the fine-grained CUB and StanfordCars, while for coarse-grained Cifar-10 and STL-10 improving the state of the art. [119]. Elastic augmented convolutions (Chapter 3) and elastic natural perturbed training generally perform the best among various natural perturbations.

5

LEARNING ATTRIBUTES FOR EXPLAINABLE ROBUSTNESS

5.1 INTRODUCTION

Explainability and robustness are key to the deployment of computer vision in the real world. In current practice, the presence of noise, occlusion, and blur easily derails a machine classifier [76], [30], [31]. Humans can generalize in the presence of perturbations; therefore, they also expect classifiers to perform well when the image is less than perfect. In other words, robustness is needed to build trust in the classifier's outcome. Furthermore, explainability is needed when the circumstances are deviating, so the user can build an understanding of why and when the classifier went off track. We conclude it is natural to combine robustness and explainability, even at the expense of losing a few percent of the classification accuracy, as one gets trust in return [29].

To achieve explainable robustness, we take inspiration from the way humans discuss visual classification. Birds are discriminated by the color of their beak, stripes on their wings, and other assorted attributes, present or absent. In microscopical pathology and radiology, trainees are pointed at visual abnormalities named by their texture. Visual classification is explained by pointing out localized attributes. A *New-England house with a classic pillar front* style classification depends on localized attributes. In the search for a missing person, *white hair and a thorn t-shirt* makes clear where and what to look for. In this paper, we propose to learn localized attributes, providing robustness and visual explanation against perturbations in the input. Hence, we aim to achieve a gain in trust at an acceptable, small loss in classification accuracy.

Attributes can be defined at the image-level [88, 100], as class-level descriptions [152], [130], or be found automatically [124]. Attributes have been used as a concise high-level descriptive or discriminative summary of an image [53], [54], [39], [44], [43]. And, they facilitate the transfer of knowledge in zero-shot classification [5], [6], [5], [141]. More than the above definitions we aim to focus on localized attributes, making it easy to use in visual explanations. We propose a new network architecture while defining *attributes* as localizable and persistent visual properties shared among the members of the same class. Not all attributes will be unique, as we expect some of them to be out of view or not detected, but the ensemble will deliver the classification. In learning attributes, we avoid image-level annotation of attributes [88] as it is very, very costly. Rather we either employ the same human-defined class-level attributes applied to all members of the class, or we use no human knowledge on attributes at all and perform machine-learned attributes.

Early work on robustness focused on adversarial perturbations to fool classifiers [51], [122], [91] with minimum impact on the image. However, later it became clear that this

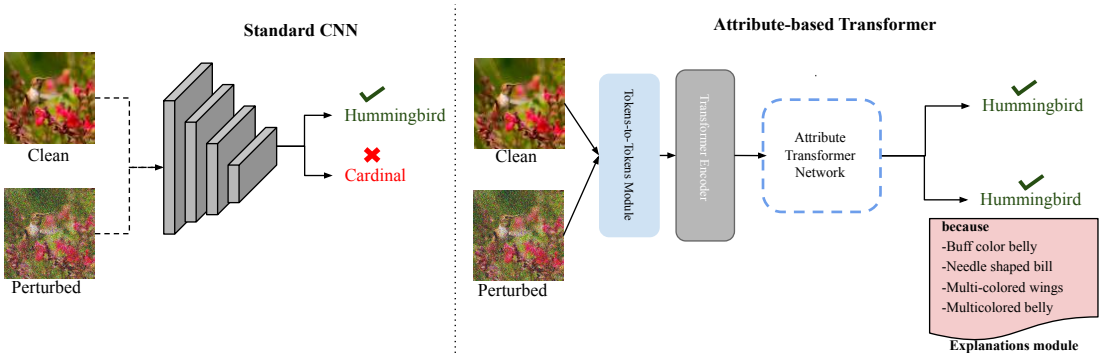


Figure 35: A standard neural network (left) and our attribute based transformer (right) at test time for classification. The attribute transformer has integrated attributes, which allow for better robustness against perturbations along with human understandable explanations.

approach does not guarantee general robustness [147], [38], [55]. As a consequence, extensive work was done on perturbation benchmarks, opening up the possibility of evaluating networks against more natural perturbations [63]. We will use this benchmark test, while focusing on the hard case of *general* robustness where perturbations are *not seen* during training.

Vision transformers [127], [35], [145] have improved computer vision classification considerably [127], as well as object detection [18], image segmentation [132], and video segmentation [133]. The key to success is the build-in self-attention mechanism, which models long-range dependencies of input image patches [142], while it is less good at capturing local dependencies [145]. In our definition of attributes, we need to cover all localized aspects, whether long-range or short-range relations, so we modify the design of transformers to cover both of them.

We make the following contributions:

- We propose a new attribute-based transformer architecture to incorporate concise attribute descriptors.
- We provide built-in visual and verbal explanations of classification decisions aligned with human visual instruction.
- We evaluate the effectiveness of human-specified attributes by class-annotation against query-specified attributes requiring no further annotation.
- We demonstrate a considerable gain in *general* robustness on natural perturbations *not seen* during training, while also providing visual explanations.

5.2 RELATED WORK

Attributes. Attributes have been used for classification [44], [39], [131], [92], for instance search [124], [77], [41], [113], for object description [39], facial recognition [23], and for inter-category transfer and zero-shot learning [83], [6], [5], [141]. In our previous work, we have used them to provide counterfactual explanations [53], [54]. Among these works, there is little consensus on the definition of attributes. The common way

to define attributes is as a human-nameable properties [35]. We prefer to use a more precise definition of attributes as *visually localized and identifying properties of an object*. Localization may refer to a part of the object, *has wing*, or to the object as a whole, *glossy surface*, that is to a bounded region. Identifying refers to contribution of the attribute to the class membership by shape, texture or color, *spotted wing*, *stripe-like headlights*, *red beak*. It allows the description to be discriminating against other classes like *has spotted wing*.

Attributes have been learned by using expert annotations [39], [44] per image. This requires enormous efforts of annotation, exceeding the usual class label per image by a factor 10 to 100 as every attribute has to be checked on visual presence in this particular image. The effort obstructs the use of large sets of attributes or large data sets. In our work, we feed a class-level attribute descriptions into the self-attention mechanism of the transformer architecture [127], [35], [145]; that is we use the same attribute descriptions for all members of the class. As an alternative, inspired by the work on self-selecting attributes [144], [124], we introduce attribute queries in the transformer architecture to learn attributes automatically. We compare the effectiveness of the two approaches.

Robustness to Natural Perturbations. Besides the susceptibility of neural networks to adversarial perturbations [93] [111], they have been shown to be vulnerable to natural perturbations such as noise or blur [31], low quality image formation [76], [29], [30], or small translations and rotations [38]. To gain public confidence, our long-term plan is to provide for near-human robustness against common perturbations.

In the literature, there are two types of robustness: *specific robustness* where the perturbations are known during the training of the classifier,and *general robustness* where the perturbations are not specifically known during the training. When training systems for robustness the result between the two types can be considerable [106]. We focus on the harder and more broader applicable case of general robustness. In [63], [62] a set of fifteen computer-generated, natural perturbations like Gaussian noise, glass blur, and elastic deformation have been collected, each with five levels of severity. These perturbations can be applied to any dataset. In this work, we use this benchmark to assess robustness while none of these disturbances was modeled explicitly.

[106] trained neural networks with tuned Gaussian and speckle noise. Similarly, [55] compared the robustness of networks trained on a variety of natural perturbations after standardizing their impact on classification. In [123], [109] normalization techniques were introduced to improve the generalization of neural networks. Rather than modifying the training of networks on internal or external perturbations, or the normalization thereof, we focus on the role of attributes to enhance robustness while being able to explain the decision when it goes wrong.

Attributes for Robustness to Natural Perturbations. In, [67] and [12], attributes are used to generate perturbed samples. And attributes are used in [53] for explaining the cause of the network's misclassification by using counterfactual explanations. In this work, we focus on utilizing attributes for enhancing the robustness of the network while providing explanations which qualitatively focus better on the location of the attribute in the image.

In [49], the authors propose an adversarial training technique that learns to generate novel images from attributes. The images are used to learn a better robustness. While the reference creates a defense against adversarial perturbations, we aim to provide general

robustness ideally applicable to *all* perturbations, or at least as broadly applicable as the range of fifteen different perturbations in the benchmark [63].

Vision Transformers for Robustness. A recent paper [13] studies the robustness of transformers against natural perturbations. They conclude that, when trained on sufficiently many data, transformers are as robust as their Resnet counterparts. We also compare the robustness of Resnet and transformers on natural perturbations and find that our transformer-based model has a significantly better robustness than Resnet.

Most similar to our work is [94], who analyzed the effect of each component of the transformer on robustness, combining those, which improve the score. They concluded that low-level patch embedding and reducing the spatial resolution while going up the layers, both enhance robustness. In this work, we also adopt low-level patch embeddings in the form of tokens, and we introduce localized attributes effectively reducing the spatial resolution in the higher layers to improve robustness. Our architecture differs from theirs primarily in the use of attributes, an approach which permits the use of class-level descriptions needed to provide visual explanations. This choice has large consequences for the network design.

5.3 METHODS

Robust classification tackles the problem of classifying images correctly in the presence of perturbations, which we aim to extend into explainable robustness. The training set consists of images x and their respective ground truth classes y.

We present two types of attribute transformer networks. In the first scenario, we use class-level K different human-descriptions specifying the possible presence of multiple attributes $\phi(y) \in \mathbb{R}^K$ and design *attribute-guided* and *attribute-embedded* networks, see Figure 36. After incorporating human-descriptions of class attributes in the network, we arrive at an alternative scenario, where we learn attributes automatically in an *auto-attribute* network.

In both cases, our network consists of three modules, the tokens-to-tokens module, an encoder, and a decoder module that learns attribute embeddings.

5.3.1 *Tokens-to-Tokens Module*

The tokens-to-tokens [145] module models the local structure, edges, and lines represented by surrounding tokens. Given an input, x the module models the local structure while reducing the length of tokens iteratively:

Restructuring. Tokens from the previous transformer layer are passed through the self-attention, normalization and multi-layer perceptron modules.

$$\hat{T} = \mathrm{MLP}(\mathrm{MSA}(T)) \tag{5.1}$$

$\hat{T}$ is then reshaped as an image in the spatial dimension.

$$x = \mathrm{reshape}(\hat{T}) \tag{5.2}$$

where, $\hat{T} \in \mathbb{R}^{l \times c}$ and $x \in \mathbb{R}^{h \times w \times c}$.

Soft Splits. To reduce the length of the tokens while preserving the local structure, a soft split [145] is applied on the restructured image. For $k \times k$ split with s overlapping and p padding, the stride is $k - s$. Each split patch has a size, $k \times k \times c$, and the output of the soft split is given to the next tokens-to-tokens process. For the input image, at first soft split is performed $T_1 = SS(x_0)$.

The procedure for the tokens-to-tokens module is summarized in Figure 36 in the blue block. It is specified as:

$$\hat{T}_i = \text{MLP}(\text{MSA}(T_i)) \tag{5.3}$$

$$x_i = \text{reshape}(\hat{T}_i) \tag{5.4}$$

$$x_{i+1} = \text{SS}(x_i) \tag{5.5}$$

The fixed-length output T_f from the tokens-to-tokens module is given to the transformer encoder f.

5.3.2 *Transformer Encoder*

The encoder f receives the fixed-length output T_f from the tokens-to-tokens module along with the position embedding.

We complement encoder input with the classification token and position embedding for attribute-guided and attribute-embedded models. The encoder layers have a standard transformer architecture with each block consisting of normalization, self-attention, normalization, and multi-layer feed-forward network. As for attribute-guided and attribute-embedded models, we use human-annotated class attributes and learn attribute queries; therefore, to extract features efficiently, we use a deep encoder with 14 blocks.

As an alternative, when one wants to avoid human-specified attribute descriptions all together, we propose automated attribute learning. There are two main differences between automated attribute learning encoder, and attribute-guided and attribute-embedded encoders that are, the number of blocks used in automated attribute learning are small, i.e., 6, and it uses single headed attention *SAttention*

5.3.3 *The Transformer Decoder*

We modify the tokens-to-tokens architecture and introduce a decoder g module. It takes the features extracted by the encoder $f(T_f)$ and attribute queries a as input. The architecture for attribute-guided and attribute-embedded decoders consists of normalization, self-attention, normalization, encoder-decoder attention, and finally, the multi-layer module. Unlike a standard transformer decoder in our network, to learn attributes for enhancing the robustness of the neural network, we introduce *attribute queries* and learn them. As for attribute-guided and attribute-embedded networks, the deep encoder architecture have already learned global and local features through class-attribute supervision therefore, we introduce a single block decoder.

For the auto-attribute learning network, we introduce a 14 block decoder because auto-attribute do not have any human annotated attribute supervision to learn attribute

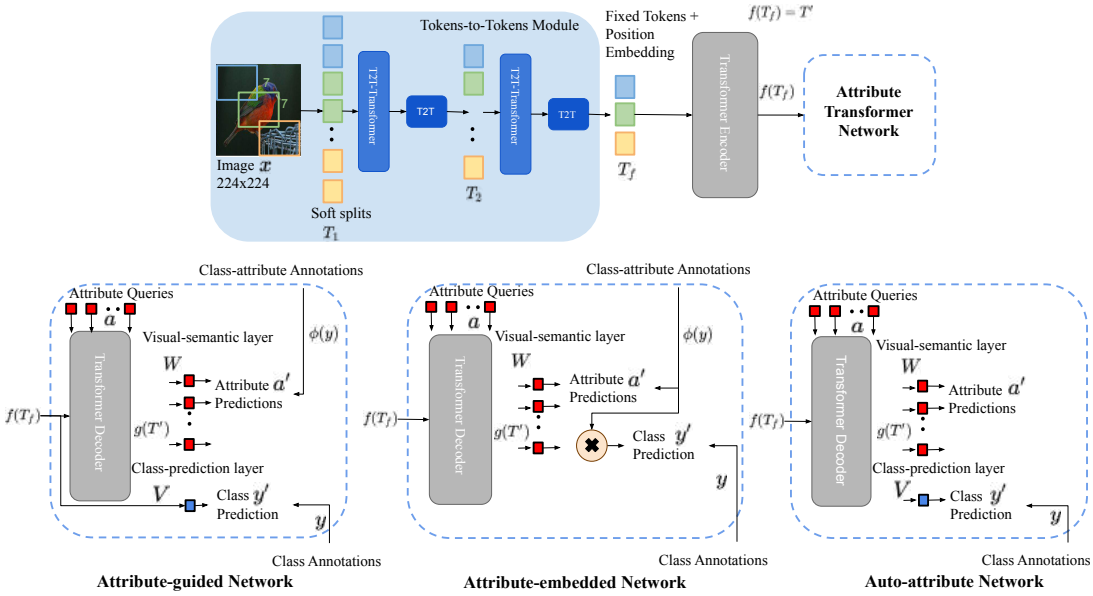


Figure 36: Our explainable attribute transformer network learns attribute queries to perform explainable robust classification. ***Attribute-guided network:*** *the decoder takes features $f(T_f)$ from the encoder and learns attribute queries a to predict attributes a'. Classes are predicted using features through class-prediction layer V.* ***Attribute-embedded network:*** *the decoder takes features from the encoder and learns attribute queries to predict per-image attributes, and perform class predictions. Classes are predicted using predicted attributes a'.* ***Auto-attribute network:*** *the decoder takes features from the encoder and learns attribute queries to predict per-image attributes, and perform class predictions. Classes are predicted using predicted attributes. It does not use any class-attribute annotations, but learns attribute queries on its own.*

queries. Similar to auto-attribute encoder, auto-attribute decoder also uses single headed attention *SAttention* instead of self-attention.

5.3.4 *The Visual Semantic Layer*

Visual semantic layer takes the visual features extracted by the decoder $g(f(T_f), a)$ and predicts per image attributes. These attributes are further used to perform classification and provide explanations.

Attribute-guided Network. For the attribute-guided network, we give visual features $g(f(T_f), a)$ extracted from the decoder to the visual-semantic layer W to predict K per-image attributes. We also map visual features extracted from the encoder $f(T_f)$ directly to the class predictions y' in the class-prediction layer V.

$$a' = g(f(T_f), a)W = g(T', a)W \tag{5.6}$$

$$y' = f(T_f)V = T'V \tag{5.7}$$

Attribute-embedded Network. For the attribute-embedded network, instead of predicting classes directly, to enhance the expressiveness of the extracted features, we first map visual features $g(f(T_f), a)$ to the attribute space and predict K attributes. Then, we

perform the dot product between the projected features and the attributes to predict the classes.

$$a' = g(f(T_f), a)W = g(T', a)W \tag{5.8}$$
$$y' = a'\phi(y)^T \tag{5.9}$$

Auto-attribute Network. For the auto-attribute network, the network learns attribute embeddings on its own. To that end, we map our visual features to two different embeddings and take a dot product between the two to predict the classes.

$$a' = g(f(T_f), a)W = g(T', a)W \tag{5.10}$$
$$z' = g(f(T_f), a)V = g(T', a)U \tag{5.11}$$
$$y' = \frac{(z')^T a'}{\sqrt{N}} \tag{5.12}$$

where N is the total number of classes.
Loss. We consider attribute predictions as a regression task and minimize the mean square error between class-attributes $\phi(y)$ and the predicted attributes:

$$\mathcal{L}_{attr} = \|a' - \phi(y)\|_2^2 \tag{5.13}$$

For the class prediction we use a cross-entropy loss enforcing the most probable class to have the highest probability:

$$\mathcal{L}_{cls} = \frac{-\exp(y'_i)}{\sum_j^N \exp(y'_j)} \tag{5.14}$$

The total loss to train the network is given as:

$$\mathcal{L} = \mathcal{L}_{cls} + \mathcal{L}_{attr} \tag{5.15}$$

5.4 EXPERIMENTS AND RESULTS

In this section, we perform experiments on three datasets of different size and granularity. On all datasets, we analyze the performance on clean and perturbed images. And, we present qualitative evaluations of our method for both clean and perturbed inputs.
Generating Natural Perturbations. We use the extensive benchmark generator for natural perturbations [63]. These perturbations consist of 18 computer-generated patterns, each with five severity levels, leading to 90 perturbations in total. We apply these corruptions to all the three datasets to generate perturbed samples for our evaluation.
Datasets. We experiment on three datasets, Caltech-UCSD-birds dataset (CUB) [130], animals with attributes (AWA) [83] and Large attribute dataset (LAD) [152]. The CUB dataset consists of 11,788 images: 5994 training and 5794 test, belonging to 200 fine-grained birds classes and 312 class-level attributes. The AwA contains, 37322: training 27805, test 9517 images with 50 classes and 85 attributes per class. The LAD dataset

Input \ Network	Resnet-50	T2T-ViT
Clean-CUB	76.67	79.46
Clean-AWA	93.10	**93.93**
Clean-LAD	89.60	90.12

Table 9: Classification accuracy on clean images. In comparison with Resnet, T2T-ViT show better performance on clean inputs for CUB and LAD datasets, for AWA the accuracy both networks perform equally well.

Input \ Network	Resnet-50 attribute-guided	Resnet-50 attribute-embedded	T2T-ViT attribute-guided	T2T-ViT Attribute Embedded	T2T-ViT Auto-attribute
Clean-CUB	74.64	69.20	**79.51**	71.29	76.06
Clean-AWA	93.32	92.19	92.35	91.19	**93.74**
Clean-LAD	87.20	84.78	**90.65**	88.09	89.02

Table 10: Classification accuracy on clean images. In comparison with Resnet-attribute guided and Resnet-attribute embedded T2T-ViT attribute variats show better performance on clean inputs for CUB and LAD datasets, for AWA the accuracy both networks perform equally well.

contains 78017 images with 54610 training and 23407 test images belonging to 230 classes and 359 attributes per class. LAD is a fine-grained dataset with five coarse-grained super categories that are further split into fine-grained classes.

Implementation Details. We take an Imagenet pre-trained Tokens-to-tokens-14 (T2T-ViT-14) transformer [145] as our baseline and fine-tune it for each dataset. We also adopt the Tokens-to-tokens architecture and pre-train it on Imagenet, and then train it on our respective datasets. We use an SGD optimizer with a learning rate of 1e-3, momentum 0.9, and weight decay 0.0005. We use the cosine annealing learning rate scheduler with a minimum learning set at 5e-5.

We employ Resnet-50 based architectures for comparison. We select Resnet-50 because it is comparable with Tokens-to-tokens-14 transformer in terms of number parameters, i.e., 25.5M, while the Tokens-to-tokens-14 transformer has 25.5M parameters. The Resnet-50 is also pre-trained on Imagenet and fine-tuned on the respective datasets. We use an SGD optimizer with a learning rate of 1e-3, momentum 0.9, and weight decay 0. We use the step learning rate scheduler with gamma set at 0.1 and step size 20. For both the Resnet and transformer networks, we use an image input size of, 224×224. The only data augmentation used during training is a random horizontal flip.

5.4.1 *Baseline Network Evaluation*

Evaluating on Clean Images

We begin by training the baseline, Imagenet pre-trained, Tokens-to-tokens transformer on each dataset. We achieve an accuracy of 79.46%, 93.93% and 90.12% on the clean images of the CUB, AWA and LAD datasets, respectively. We also train the Imagenet pre-trained Resnet-50 network on each dataset of clean images. It gains 76.67%, 93.10% and 89.60% on CUB, AWA and LAD datasets, respectively, see Table 9.

Input \ Network	Resnet-50	T2T-ViT	Resnet-50 attribute-guided	Resnet-50 attribute-embedded	T2T-ViT attribute-guided	T2T-ViT attribute-embedded	T2T-ViT Auto-attribute
Gaussian Noise	57.38	67.89	56.08	51.65	70.17	61.35	63.23
Shot Noise	55.29	67.60	54.28	49.93	70.03	61.29	63.20
Impulse Noise	32.36	60.41	31.46	26.83	62.72	53.38	54.84
Defocus Blur	73.87	76.82	74.05	65.87	77.45	68.68	72.94
Glass Blur	50.86	63.38	51.20	42.78	63.10	58.19	56.35
Motion Blur	64.17	69.89	64.12	55.81	69.88	60.91	64.63
Zoom Blur	55.07	60.22	54.87	47.30	57.45	51.36	52.40
Snow	73.18	77.47	72.63	64.30	77.05	68.85	74.05
Frost	66.44	72.11	66.53	57.93	72.33	63.19	67.18
Fog	71.61	78.16	71.51	62.43	78.20	70.13	75.00
Brightness	74.80	77.19	74.14	65.13	77.47	69.15	74.49
Contrast	61.46	75.56	60.80	51.17	77.09	66.63	70.92
Elastic Transform	74.67	77.00	74.34	66.03	77.37	69.35	73.26
Pixelate	70.56	74.35	70.76	62.59	73.88	67.07	70.11
Jpeg Compression	71.34	73.12	71.12	62.60	69.20	65.70	67.60
Speckle Noise	53.87	66.71	52.77	47.80	69.44	60.42	63.80
Gaussian Blur	72.75	75.76	72.99	64.71	76.48	67.54	71.43
Saturate	57.03	63.15	54.11	47.30	69.90	54.09	63.92
Average	63.15	70.93	62.65	55.12	**71.62**	63.18	66.63

Table 11: Classification accuracy on perturbed images for CUB dataset. In comparison with Resnet T2T-ViT and its variations show better robustness against perturbed inputs with the T2T-ViT attribute guided being the best.

Evaluating on Perturbed Images

To evaluate the robustness of our Resnet and T2T-ViT baseline networks, on the benchmark [63], we average the results of the five severity levels. They are shown in Table 11, 12 and 13 (column 2 and column 3) for the CUB, AWA and LAD datasets, respectively.

From the Tables 11,12 and 13 it is clear that both baselines are far from being robust against the natural perturbations. The performance over clean image tests in Table10 drop by 13.52%, 8.82% and 11.95% respectively for Resnet. However, T2T-ViT is more robust for all datasets, than the Resnet-50 model, but still its performance decreases by 8.53%, 5.73% and 8.14%. The attention mechanism in T2T-ViT, which connects each pixel of the image to every other pixel, shows more inherent robustness against perturbations. This motivates us to proceed with this model to further enhance its robustness by introducing attributes in the network for explainability, even when it would have a modest negative impact on the classification performance.

5.4.2 *Evaluating Attribute Networks*

Evaluation on Clean Images

We incorporate class attributes in both the Resnet-based and transformer-based networks. We consider two Resnet-based networks and three transformer-based networks. The results for clean images are presented in Table 10.

We observe that by introducing human-described class attributes into the network through attribute-guided training, the performance for the Resnet-50 drops compared to the baseline network without attributes. The same is true for the attribute-embedded training of Resnet. This reduction is because attributes restrict the focus for the network, i.e., with localized attributes the network focuses on the salient parts of the object rather than the whole scene. The information in the background is removed, which will, in general, assist to improve the classification. For T2T-ViT-based models, we observe that the networks, especially the attribute-guided network, keeps up very well with the

Input \ Network	Resnet-50	T2T-ViT	Resnet-50 attribute-guided	Resnet-50 attribute-embedded	T2T-ViT attribute-guided	T2T-ViT attribute-embedded	T2T-ViT Auto-attribute
Gaussian Noise	84.13	86.73	83.54	81.03	79.41	77.27	87.5
Shot Noise	84.86	86.73	84.41	82.07	80.41	78.21	87.72
Impulse Noise	64.46	82.43	64.22	59.85	69.78	67.77	83.02
Defocus Blur	91.7	93.03	91.77	90.58	90.11	88.82	92.74
Glass Blur	73.29	78.56	74.35	71.58	76.74	74.95	80.26
Motion Blur	80.06	88.36	80.37	77.86	81.81	81.94	85.27
Zoom Blur	57.51	62.51	56.44	56.65	60.40	62.14	60.45
Snow	91.3	93.56	91.25	90.15	91.40	90.19	93.02
Frost	85.64	89.27	85.43	83.57	85.30	83.16	88.61
Fog	90.38	93.24	90.05	89.19	91.52	90.28	93.13
Brightness	92.08	93.42	92.01	91.03	91.48	90.18	93.02
Contrast	84.08	92.61	83.67	82.76	87.96	86.17	92.18
Elastic Transform	91.63	93.21	91.60	90.58	90.27	89.05	92.52
Pixelate	89.41	90.65	89.29	87.41	88.58	88.10	90.15
Jpeg Compression	90.77	91.53	90.81	89.50	90.05	88.34	90.17
Speckle Noise	84.64	86.94	84.03	82.00	81.04	78.87	88.08
Gaussian Blur	91.17	92.6	91.16	89.92	88.96	87.67	92.24
Saturate	89.89	92.29	89.69	88.37	87.59	86.17	91.88
Average	84.28	**88.20**	84.12	82.45	84.05	82.74	**87.89**

Table 12: Classification accuracy on perturbed images for AWA dataset. In comparison with Resnet T2T-ViT and its variations show better robustness against perturbed inputs with the T2T-ViT and T2T-ViT auto-attribute guided being the best.

accuracy, i.e., 79.51 %, 92.35 % and 90.65 % for the clean CUB, AWA and LAD datasets, respectively. This is considerable as the attributes are a restriction in the optimization of the network which delivers a visual explanation, but a constraint in the optimization nevertheless.

Evaluation on Perturbed Images

After incorporating human-described class attributes in the network, we also test them on perturbed inputs at five levels of severity.

The results for the CUB-dataset are shown in Table 11. We observe that the Resnet-50 attribute-guided network maintains the robustness against perturbations, while the attribute-embedding network drops in performance from 63.15% to 55.12 %. On the other hand, incorporating attributes in the network leads to a small improvement in the performance against perturbations for the CUB birds dataset from 70.93% to 71.62%. Our auto-attribute network maintains its performance and performs equally well as the original T2T-ViT transformer network.

AWA dataset results in Table 12 shows that auto-attribute and original T2T-ViT transformer perform the best. Different from CUB dataset where incorporating attributes through attribute-guided network enhanced robustness, here auto-attribute performs better. We attribute this behavior to the number of human annotated attributes available with the datasets. As in AWA dataset the number of attributes are only 85 attributes per class therefore it does not enhance the robustness.

For the LAD dataset, Table 13 with class-attribute information in the network, Resnet-50 shows a robustness similar to the standard network without attributes. However, T2T-ViT demonstrates a significant improvement in the robustness against perturbations from 81.98% to 84.45 %. Hence, by incorporating class attributes in the network, the robustness of the transformer network against perturbations improves.

Input \ Network	Resnet-50	T2T-ViT	Resnet-50 attribute-guided	Resnet-50 attribute-embedded	T2T-ViT attribute-guided	T2T-ViT attribute-embedded	T2T-ViT Auto-attribute
Gaussian Noise	74.25	78.24	74.21	70.96	82.83	73.94	75.61
Shot Noise	72.15	77.84	71.89	68.98	81.76	73.33	75.35
Impulse Noise	53.52	73.72	51.78	47.88	79.61	66.00	69.70
Defocus Blur	88.04	88.30	87.63	85.38	89.59	86.08	87.24
Glass Blur	49.54	63.39	48.82	44.59	74.29	60.98	61.10
Motion Blur	78.05	80.13	77.23	73.95	82.51	78.49	77.92
Zoom Blur	63.21	64.27	62.31	57.70	66.11	61.70	59.51
Snow	86.54	89.19	86.42	83.57	89.63	86.56	87.95
Frost	76.75	82.37	76.29	72.67	83.62	78.86	79.14
Fog	86.76	89.29	86.55	84.18	90.05	87.61	88.19
Brightness	88.00	89.06	87.58	85.13	89.58	86.65	87.87
Contrast	81.22	88.01	80.35	77.79	89.16	84.67	86.40
Elastic Transform	87.52	88.34	87.25	84.86	89.17	86.26	87.27
Pixelate	84.62	85.96	84.15	81.19	87.23	84.99	85.42
Jpeg Compression	86.89	87.00	86.40	84.23	87.96	85.78	85.76
Speckle Noise	70.26	77.01	69.67	66.77	80.50	72.22	74.77
Gaussian Blur	87.32	87.45	86.94	84.59	88.99	85.06	86.26
Saturate	83.13	86.05	82.85	80.09	87.45	82.28	84.19
Average	77.65	81.98	77.13	74.14	**84.45**	78.97	79.98

Table 13: Classification accuracy on perturbed images for LAD dataset. In comparison with Resnet T2T-ViT and its variations show better robustness against perturbed inputs with the T2T-ViT attribute guided being the best.

5.4.3 *Explanations using Attributes*

We select our T2T-ViT attribute guided network, generally showing the best clean and perturbed accuracy, to generate explanations for the classification decisions of the network.

Explanation for correct classification

Figure 37 shows some qualitative examples for explanations provided by our T2T-ViT attribute guided network. We observe that when a clean input is correctly classified into its respective class, the predicted attributes for that image align with the correct class. For example, the second image in the figure is classified into “Glaucous winged Gull” and the attributes predicted for this image tell us why it is classified into Glaucous winged Gull class because it has Grey back, Grey upper tail and Grey under tail clearly visible in the image. Similarly, the fifth image is classified into “Painted bunting class” because of the attributes orange belly, multicolored wings, multicolored breast etc. Hence, our attribute predictions provide human understandable reasoning behind the classification decisions.

Explanation for misclassification

In Figure 38 we show some qualitative examples of explanations provided by our T2T-ViT attribute guided network for correct classification of clean input and misclassification of the perturbed input. The first row shows shot noise perturbed inputs and the second row shows zoom blurred inputs.

We see that, when a clean input is classified into the correct class, its attributes align with the respective class. While, when the perturbed version of the same image gets misclassified into the wrong class, its attributes start indicating towards the wrong class providing the reasoning behind why it got misclassified. For instance, in the figure, first image in the second row when it is classified correctly to the “Black footed albatross” its

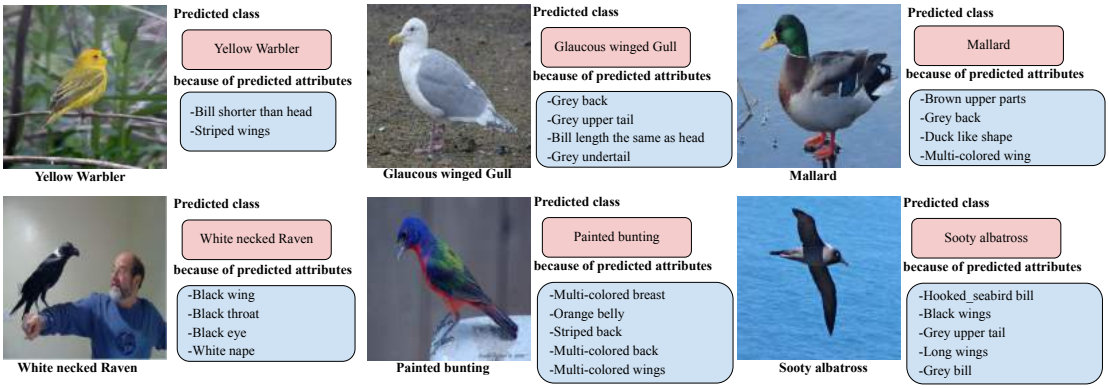


Figure 37: Qualitative examples of our attribute-based explanations for clean images from the CUB Dataset. We observe that our human understandable predicted attributes align with the class predictions and tell us why a specific image got classified into a particular class.

attributes are hooked seabird bill, buff color belly clearly visible in the image. However, when perturbed it got misclassified into "Elegant tern" class the attributes values for hooked seabird bill and buff color belly decreased and multicolored wings value increased which lead to misclassification. Likewise, for the second image in the second row when the predicted attributes were buff color belly, multicolor belly and striped back it got classified as "Rufous humming bird" however when their values decreased it got misclassified into "Rubythroated humming bird" class. Hence, our attribute based explanations provide the reasoning behind when a clean image gets classified correctly and when a perturbed image gets misclassified.

5.5 CONCLUSION

We have defined attributes as localized and identifying characteristics of an object class. The localized and identifying components in the definition of attributes, are directly translated into the components of a new transformer architecture to incorporate these attribute descriptors.

One version of our architecture implements the human-specified class-level attributes as queries to the transformer. The alternative version of our architecture does not use human-specified description, but rather generates localized and identifying attributes itself for our main purpose of providing visually explained robust classification. We compare these two architectures with a Resnet architecture. We conclude that our transformer-based architectures are much, much more robust against perturbations than the Resnet architecture, while also the visual explanation they provide is substantially better than resulting from the alternative architecture. The comparison between two transformer networks indicates that the proposed network with human-specified class-attributes performs slightly to moderately better as expected as more a priori knowledge is used, but it has surprised us how well the auto-learning capacity of the attribute-transformer network was capable of keeping up in robustness.

Our attribute-based visual explanations provide us the reasoning behind why a clean input without inflicted perturbations get classified correctly and why the perturbed input

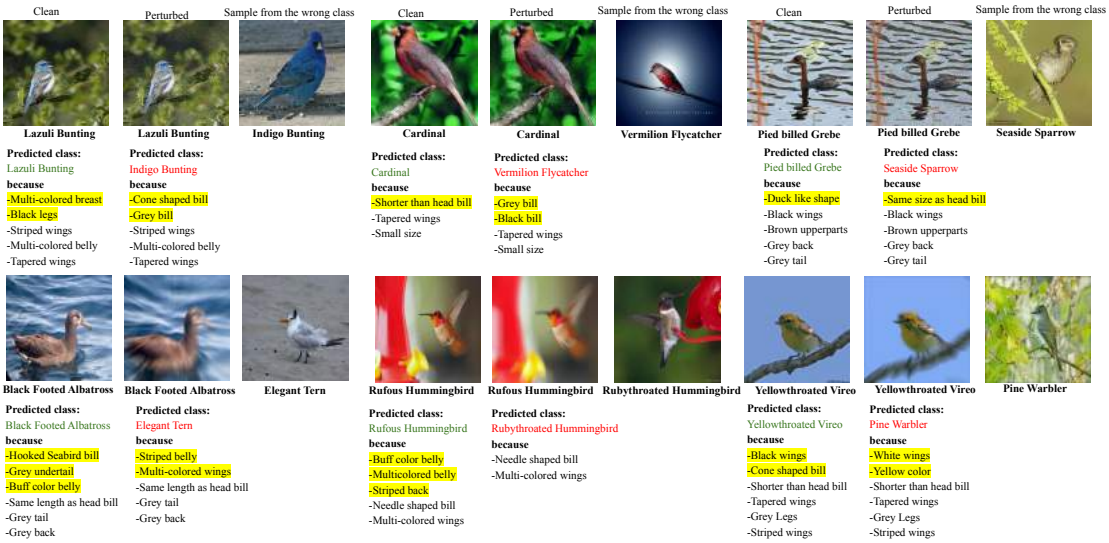


Figure 38: Qualitative examples of our attribute-based explanations for perturbed (Row 1 shot noise, Row 2 zoom blur) images from the CUB Dataset. Our human understandable predicted attributes tell us why a specific image got misclassified into a particular class. Highlighted attributes for the clean images are related to correct classes, while those for perturbed images align with the incorrect class. Those non highlighted ones are common among both classes.

get classified into the wrong class. Where it is hard to ascribe the definitive proof to qualitative examples, the failure cases highlighted in Figure 38 all clearly indicate the detected presence of false attributes, that is attributes which are not identifying for the true class. In that sense, our attribute-transformer network provides visual explanations of correct and incorrect classification.

6

SUMMARY AND CONCLUSIONS

6.1 SUMMARY

In this thesis, we explored the explainable robustness of neural networks for visual classification. It began with enabling black-box neural networks to justify their reasoning by leveraging attributes, i.e., visually discriminative properties of objects, and perturbations, to provide counterfactual explanations. The two chapters that followed focused on enhancing the robustness of neural networks against natural and adversarial perturbations. We did so by integrating perturbations in the network architecture and provided a rationale behind the modification of the network for enhancing its robustness by training the standard network with similarly transformed images. The last chapter utilized attributes to improve robustness against perturbations and provided explanations as a byproduct.

In Chapter 2, we aimed to explain the decisions of neural networks by utilizing multimodal information. That is counter-intuitive attributes and counter visual examples which appear when perturbed samples are introduced. Unlike previous work on interpreting decisions using saliency maps, text, or visual patches, we proposed using attributes and counter-attributes, examples, and counterexamples as part of the visual explanations. When humans explain visual decisions, they tend to do so by providing attributes and examples. Hence, inspired by human explanations, in this chapter, we provided attribute-based and example-based explanations. Moreover, humans also tend to explain their visual decisions by adding counter-attributes and counterexamples to explain what is not seen. We introduce directed perturbations in the examples to observe which attribute values change when classifying the samples into the counter classes. This delivers intuitive counter-attributes and counterexamples. Our experiments with both coarse and fine-grained datasets showed that attributes provide discriminating and human-understandable intuitive and counter-intuitive explanations.

Chapter 3 emphasized the importance of robustness against unwanted perturbations. Robustness against perturbations like noise, blur, saturation, and occlusion is essential before deploying neural network classifiers in the real world. While many approaches for robustness train the network by providing augmented data to the network, we aimed to integrate perturbations in the network to achieve improved and more general robustness. To that end, we proposed a new well-founded method to deform weights by fourfold stochastic elastic deformations on the basis functions as an approximation to the effect of local perturbations. In an extensive experimental evaluation, we compared the effectiveness of locally transformed weights with globally transformed weights and found that the local ones are more robust. Then we evaluated our method on unseen

perturbations (occlusion, snow, Gaussian noise, Gaussian blur). On perturbed CIFAR-10 images, the modified network delivered better performance than the original network. For the much smaller STL-10 dataset, in addition to delivering better general robustness, wiggling even substantially improved the classification of unperturbed, clean images. We found that the robustness of the new network is further improved when combined with data augmentation. The robustification of the network comes at only limited computational costs. We conclude that locally wiggled-weight networks acquire good robustness even for perturbations not seen during training.

In Chapter 4, we proposed natural perturbed training to enhance robustness besides providing the reasoning behind the effectiveness of integrating natural perturbations in the network, proposed in Chapter 3. To develop a relationship between input image transforms and their respective filter transforms, natural perturbed training was introduced, in which we trained the network on perturbed inputs. Natural perturbations are encountered in practice: the difference of two images of the same object may be approximated by an elastic deformation (when they have slightly different viewing angles), by occlusions (when they hide differently behind objects) or by saturation, Gaussian noise, etc. Training some fraction of the epochs on random versions of such variations will help the classifier to learn better. We conducted extensive experiments on six datasets of varying sizes and granularity. Natural perturbed training showed better and much faster performance than adversarial training on clean, adversarial as well as natural perturbed images. It even improved general robustness on perturbations not seen during the training. For CIFAR-10 and STL-10, natural perturbed training even improved the accuracy for clean data and reached state-of-the-art performance. Ablation studies verified the effectiveness of natural perturbed training. Among various natural perturbations, elastic perturbed training performed the best. This can be understood from the fact that an elastic transformation corresponds to small deviations in viewpoint, hence filling in one of the main sources of variance in the network architecture. From the experiments, we conclude that our natural perturbed training obtains a good level of general robustness.

Explainability and robustness are key to the employment of computer vision in the real world. In chapter 5, we proposed the use of integrated localized attributes against perturbations in the input. A novel transformer network learned discriminative local attributes to enhance the robustness and provide explanations for classification. While we did not need attribute annotations per image, we considered two efficient levels of attribute knowledge: incorporating class-level descriptions, like “seagulls have yellow beaks” to all class members. Furthermore, the other was to use no human description, but rather let the system learn the attributes by asking queries. Both provided explanations similar to the way humans would use to relate visual classification decisions. We demonstrated that the new architecture improves the robustness of classification systems and provides visual explanations at the expense of some loss in accuracy.

6.2 CONCLUSIONS

In this thesis, we study an essential question for making neural networks deployable in real-world applications: *how to make neural networks explainably robust?* We start by making neural networks explainable. In order to provide human-understandable explana-

tions in perfect scenarios (clean inputs) and for imperfect scenarios (perturbed inputs), we utilize attributes and ask: *can an explainability model provide factual and counterfactual explanations?* Our results show that attributes provide human-understandable explanations, and they are crucial for the discrimination between classes. We also show that for adversarially perturbed images (images with deliberate distortions causing a minimal change in the appearance but a maximal change in the classification), attributes indicate to the class in which the image is misclassified, and when a network is robustified against such adversarial perturbations, the predicted attribute values for the perturbed images start indicating back towards the correct class, which further confirm our attribute-based explanations. Hence, we conclude that attributes provide intuitive factual and, in the presence of perturbations counterfactual human-understandable explanations. These explanations could also be enhanced by retrieving visual examples through them. Attributes retain their best discriminative power in the presence of perturbed inputs with standard and robustified networks.

Although perturbations could be utilized to generate counterfactual explanations, it is equally important to make networks robust against perturbations, therefore in chapter 3 we ask: *how to integrate perturbations into the network to enhance its general robustness?* We formulate a method to enhance the robustness of networks for classification against common perturbations. Our results demonstrate the effectiveness of our method by improving the performance against natural and adversarial perturbations over standard networks. Local elastic convolutions corresponding to viewpoint change deformation generally perform the best. We conclude that our wiggled weights approach induces good general robustness for the class of such natural perturbations.

After achieving general robustness through integrating perturbations in the network in chapter 4 we build a connection between builtin transformations and input image transformations and ask: *how to train a neural network on perturbations to enhance its general robustness?* We provide a rationale behind the modification of the network for enhancing its robustness by training the standard network with similarly transformed images. Our results show that natural perturbed training, while being computationally fast, also shows better generalization on adversarial and natural perturbations (Gaussian noise, blur, snow) than adversarial training. Moreover, it improves the classifier accuracy on clean images. Elastic augmented convolutions (chapter 3) and elastic natural perturbed training (chapter 4) generally perform the best among various natural perturbations.

Our results from chapter 2 till 4 show the importance and effectiveness of explainability, robustness, and explainable robustness. Therefore, in the final chapter of the thesis, we focus on utilizing explanations for enhancing the robustness and ask: *can attributes enhance the robustness of neural networks besides providing explanations?* We begin by defining attributes as localized and identifying characteristics of an object class. Next, we translate these attributes directly into the components of a new transformer architecture. We conclude that our transformer-based architectures are much more robust against perturbations than the Resnet architecture, while also providing the visual explanation. With human-specified attributes we would expect a small loss in the performance as by providing attributes we constrain the network's optimization however, the proposed performed slightly to moderately better. For the network without human-specified attributes, results surprised us how well the auto-learning capacity of the attribute-transformer network was capable of keeping up in robustness. Our attribute-based

visual explanations provided the reasoning behind why a clean input without inflicted perturbations got classified correctly, and why the perturbed input got classified into the wrong class.

In this thesis, we provide explanations both in perfect scenarios (clean inputs) as well as in imperfect scenarios (perturbed inputs). The common way to improve robustness is by feeding the network with perturbed data, either natural or adversarial [51, 65, 107]. We show a substantial improvement in the robustness of classifiers without any data augmentation.

General robustness, i.e., robustness against perturbations not seen during the training of the network is important, but difficult to achieve as the network needs to be robust against a wide range of unseen natural perturbations. We show a significant enhancement in general robustness by training on one type of perturbation and testing on several unseen perturbations. Different from previous works [38, 150] where authors show that enhancing the robustness of neural networks against perturbations leads to a degradation in the clean accuracy, our methods besides improving robustness against perturbations also lead to an improvement in the clean image accuracy. Furthermore, we note that although adversarial perturbations and adversarial training are intriguing, however, they do not accomplish robustness against natural perturbations, while our built-in natural perturbations as well as natural perturbed training improve robustness against adversarial perturbations. Finally, we integrated human-specified attributes in the network to provide human-understandable explanations. As integrating attributes act as a constraint for the optimization of the system, therefore, it leads to a small loss in the performance.

This thesis contributes in constructing explainably robust models. Both explanations and robustness stand alone and in conjunction find their significance in intelligent systems. To generate explanations like humans or to be robust like human visual system requires a lot of work. For example, although perturbations considered in this thesis are called natural perturbations, they only mimic natural perturbations as in fact they are generated using a computer program. The next step in this research should be making networks robust against more realistic natural perturbations, e.g., snow as in the real world. We believe that the work in this thesis will hold against those natural perturbations too.

Furthermore, we focus on automatically generating human-understandable attributes for providing explanations. This topic also requires further research to better learn human-understandable automatic attributes.

Our results showed that incorporating viewpoint changes in the neural networks in the form of elastic deformations enhances robustness. Introducing elastic deformations for video datasets is a straightforward extension of this work because in videos angles or viewpoints change between frames.

We close this thesis by noting that explainable robustness is essential for making neural networks more practical in nature, and this thesis is a small step towards it. Automatically generated human-understandable explanations, robustness against more natural perturbations, and extension to other data formats are just a few examples of making neural networks more practical.

SAMENVATTING

In deze thesis hebben wij uitlegbare robuustheid van neurale netwerken voor visuele classificatie onderzocht. In het eerste hoofdstuk laten wij zien dat de werking van black box modellen uitgelegd kan worden door eigenschappen te gebruiken zoals visueel kenmerkende objecten of verstoringen als een tegen-feitelijke uitleg. In de twee volgende hoofdstukken focussen wij op verbetering van de robuustheid van neurale netwerken tegen natuurlijke en gecreëerde verstoringen. Dit hebben wij gedaan door verstoringen te integreren in de netwerk architectuur. Daarnaast gaven wij een redenering achter de wijziging van het netwerk om de robuustheid ervan te vergroten door het standaard-netwerk te trainen met vergelijkbaar getransformeerde beelden. In het laatste hoofdstuk gebruiken wij eigenschappen om robuustheid te verbeteren tegen verstoringen en geven uitleg hierover als een bijproduct.

In hoofdstuk 2 richten wij ons op het uitleggen van beslissingen van neurale netwerken door het gebruik van multimodale informatie. Meer specifiek contra-intuïtive eigenschappen en contra-visuele voorbeelden die verschijnen wanneer verstoorde data geïntroduceerd wordt. Eerdere onderzoeken leggen beslissingen uit met tekst of visuele regio's. In tegenstelling tot dat introduceren wij het gebruik van eigenschappen en tegen-eigenschappen, en voorbeelden en tegen-voorbeelden om visuele uitleg te bewerkstelligen. Als mensen visuele beslissingen uitleggen doen zij dat vaak door voorbeelden te geven, en specifieke eigenschappen te benoemen. In deze op menselijke handelen gebaseerde methode introduceren wij eigenschap- en voorbeeld-gebaseerde uitleg. Verder legt men visuele beslissingen ook vaak uit door een tegen-voorbeeld, of tegen-eigenschap te benoemen om uit te leggen wat er niet te zien is. Daarom introduceren wij gerichte verstoringen om te zien welke eigenschappen veranderen wanneer data in een andere klasse wordt geclassificeerd. hiermee creëren wij intuïtive tegen-voorbeelden en tegen-eigenschappen. Onze experimenten met zowel grove als fijnkorrelige data sets geven onderscheidende en voor mensen begrijpelijke intuïtive en contra-intuïtive uitleg.

Hoofdstuk 3 benadrukte het belang van robuustheid tegen ongewenste verstoringen. Robuustheid tegen verstoringen zoals ruis, verzadiging en occlusie is essentieel voordat classificatie modellen gebaseerd op neurale netwerken in de echte wereld kunnen worden ingezet. Terwijl veel methodes voor robuustheid het netwerk trainen door het netwerk te voorzien van meer data, streefden wij ernaar verstoringen in het netwerk te integreren om een verbeterde en meer algemene robuustheid te bereiken. Daarom hebben wij een nieuwe methode voorgesteld om parameters te vervormen door middel van viervoudige stochastische elastische vervormingen van de basisfuncties als benadering van het effect van lokale verstoringen. In een uitgebreide experimentele evaluatie hebben wij de doeltreffendheid van lokaal getransformeerde parameters vergeleken met globaal getransformeerde parameters en vastgesteld dat de lokale robuuster zijn. Vervolgens hebben wij onze methode geëvalueerd op ongeziene verstoringen (occlusie, (Gaussische) ruis en onscherpte). Op verstoorde CIFAR-10 beelden leverde het gewijzigde netwerk betere prestaties dan het oorspronkelijke netwerk. Voor de veel kleinere STL-10 dataset

leverde onze methode naast een betere algemene robuustheid, zelfs een aanzienlijke verbetering op van de classificatie van onverstoorde oorspronkelijke beelden. Wij hebben vastgesteld dat de robuustheid van het nieuwe netwerk verder verbetert wanneer het meer data wordt toegevoegd. Deze stap naar extra robuustheid van het netwerk gaat slechts gepaard met beperkte computationele rekenkosten. Wij concluderen dat lokaal gewogen netwerken een goede robuustheid bereiken, zelfs voor verstoringen die tijdens de training niet zijn waargenomen.

In hoofdstuk 4 hebben wij een natuurlijke verstorings-training voorgesteld om de robuustheid te verbeteren, naast de redenering achter de effectiviteit van de integratie van natuurlijke verstoringen in het netwerk, zoals voorgesteld in hoofdstuk 3. Om een verband te leggen tussen de input beeld transformaties en hun respectievelijke filter transformaties, introduceren wij natuurlijke verstoorde training, waarbij wij het netwerk trainden op verstoorde inputs. Natuurlijke verstoringen komen in de praktijk voor: het verschil tussen twee beelden van hetzelfde object kan worden benaderd door een elastische vervorming (wanneer zij enigszins verschillende kijkhoeken hebben), door occlusies (wanneer zij zich verbergen achter objecten) of door verzadiging, Gaussiaanse ruis, enz. Door een deel van de epochs te trainen op willekeurige versies van dergelijke variaties kan het classificatie model beter leren. Wij hebben uitgebreide experimenten uitgevoerd op zes data sets van uiteenlopende omvang en granulariteit. Natuurlijke verstoorde training leverde betere en veel snellere prestaties dan adversariële training op zowel schone, adversariële als natuurlijke verstoorde beelden. Het verbeterde zelfs de algemene robuustheid bij verstoringen die tijdens de training niet werden waargenomen. Voor CIFAR-10 en STL-10 verbeterde de training met natuurlijke verstoringen zelfs de nauwkeurigheid voor oorspronkelijke schone data en werden state-of-the-art prestaties bereikt. Ablatie studies hebben de doeltreffendheid van natuurlijke verstorings-training geverifieerd. Van de verschillende natuurlijke verstoringen presteerde de elastische verstoorde training het best. Dit kan worden begrepen uit het feit dat een elastische transformatie overeenkomt met kleine afwijkingen in het gezichtspunt, waardoor een van de belangrijkste bronnen van variantie in de netwerkarchitectuur wordt verklaard. Uit de experimenten concluderen wij dat onze natuurlijke verstoorde training een goed niveau van algemene robuustheid bereikt.

Uitlegbaarheid en robuustheid zijn de sleutel tot het gebruik van visuele neurale netwerken in de echte wereld. In hoofdstuk 5 laten wij zien dat geïntegreerde lokale eigenschappen gebruikt kunnen worden tegen verstoringen in model input. Een transformer netwerk was getraind om onderscheidende lokale eigenschappen en robuustheid te verbeteren en om uitleg te geven over classificaties. Ondanks dat wij geen eigenschap annotatie nodig hadden per beeld, bekeken wij twee efficiënte niveaus van eigenschap kennis: ten eerste de integratie van klasse-niveau beschrijvingen, zoals ”zeemeeuwen hebben gele snavels” voor alle klasse-leden. Ten tweede door het model geen menselijk interpreteerbare beschrijving te laten leren, maar door het model de eigenschappen te laten leren door vragen te beantwoorden. Beide gaven uitleg die lijkt op de manier waarop mensen visuele classificatie beslissingen zouden nemen. wij laten zien dat deze nieuwe model architectuur de robuustheid en visuele uitleg van classificatie systemen kan verbeteren ten koste van slechts een kleine vermindering van de nauwkeurigheid.

ACKNOWLEDGMENTS

Coming from Pakistan and going to South Korea for my masters and then moving to the Netherlands for my PhD was indeed a challenging but thrilling journey. In Amsterdam I learned to look at life from a very different perspective, contrasting with the south and east Asia and all this became possible because of the opportunity provided by my supervisor Arnold Smeulders. So, first of all I would like to thank Arnold for providing me with the opportunity to come to Amsterdam and then guiding me throughout the PhD journey.

As my supervisor, Arnold plays a very important role in my PhD journey. Along with guiding me in my research he also provided me with all the facilities to do the research. I still remember our meetings in the beginning of the PhD when I found it difficult to understand and answer his fundamental, intuitive and sharp questions, which later on shaped my way of thinking. Besides guiding me in research I also greatly appreciate all the life lessons he taught me, a few of them are "Learn to say No", and "Learn to hear No","Concretely plan next steps", "you are not shy you act as shy", "you are brave you went to Korea and now you came here", "if you can survive Amsterdam you will survive anywhere in the world", "we don't update the text but we rewrite it" and most importantly "Add articles in the text where needed" and many more. You also appointed me as the Soos chair to bring me out of my comfort zone and to make me interact with other lab members, which in the long term was very useful. I am also thankful to Arnold's wife Anneke for her hospitality during the Christmas dinners and especially for teaching me how to bake a red velvet cake.

I am grateful to my PhD defense committee members Dr. H.C. van Hoof, Prof. dr. C.G.M. Snoek, Prof. dr. T. Gevers, Dr. N.J.E. van Noord, Dr. J.C. van Gemert, Prof. dr. Noel E. o'Connor. It is a pleasure to have you all in my defense committee.

Zeynep Akata is yet another person who gave me the opportunity to do this PhD and was my co-supervisor in the beginning, she was a great help in the beginning of the PhD. I wish we had further worked together. Being a successful woman in the field she is a role model for me. I am greatly thankful to her for the short time that we worked together. I am also thankful to Max Welling for letting me be the part of the lab that he and Arnold built. Although I got the chance to meet him only through biweekly lab seminars but his comments and suggestions were always very helpful. I am also thankful to Virginie Mes and Felice Arends for making administrative stuff easier for us.

In the beginning of my PhD my first interaction was with the Delta Lab members, I really enjoyed being a part of the team with the amazingly talented people. I also enjoyed our group business trips to Bosch and time spent there. It was great to know all the members Elise van der Pol, Emile Hoogeboom, Jorn Peters, Ivan Sosnovik, Thomas Andy Keller, Victor Garcia Satorras, Artem Moskalev, Wenling Shang and Sindy Lowe. Elise was the first example of a dutch girl for me and I really learned a lot from her. Emile (a dutch guy who had been to Pakistan), Jorn (Elise's friend and good at the programming stuff), Andy (makes jokes about himself), Victor (easy to talk with), Artem (thank you

for the small discussions during the work), Sindy (a very talented girl). I would also like to thank people at Bosch for having us at BCAI, Renningen for our yearly visits.

I would like to express my gratitude to Ivan for his coordination and help during the courses that we taught together and also during our collaboration. I got to learn a lot from him especially from the theoretical perspective. I always enjoyed our discussions ranging from work, to languages, cultures, politics, religions and so on especially during the COVID times at our neighbouring cafe .

I would also like to especially thank Wendy. I am thankful to her for being such a good friend and asking me to hang out together, to eat together, go shopping together and to travel together (I greatly enjoyed our trip to Lapland and the reindeer ride) and most importantly for her suggestions and recommendations in every aspect of life.

Shihan Wang yet another person who approached me and befriended me. I still remember our long meetings in the biology common room. Whenever I was down or worried about my work she was always there to provide me with the useful suggestions. We also managed to travel together to Canary Islands. I really enjoyed that trip I wish we could travel again to Japan or South Korea. Last but not least I am thankful for her suggestions for Chinese dramas.

JiaoJiao, indeed a very friendly, and a hard working peer. I am thankful for her companionship, suggestions and collaboration. I remember in the beginning of the PhD we were neighbors and she prepared dates soup (from Xian) for me. I really loved it. Many thanks to Zenglin Shi for introducing me to the Chinese culture especially food. I enjoyed tasting diverse Chinese cuisine during our summer school in London.

Many thanks to Sarah for introducing me to another perspective of Amsterdam's life. She has always been a help whether it was a problem regarding research some other administrative issue in Netherlands or some health care related issue she always gave very useful tips. She also introduced me to a variety of activities in Amsterdam like modern dance, ballet performances, good restaurants and more.

I am grateful for William's suggestions throughout the PhD. He was a great help in understanding and adjusting to Arnold's way of supervision. I am also thankful to him for realizing my interests in Asian culture and then inviting me to the related activities.

Interestingly I learned about China and Chinese culture, food and language in Amsterdam more than my sister did in Beijing, that is all because of my Chinese colleagues at the VIS lab. I am thankful to all of them Zenglin Shi, Shuo Chen, Yunlu Chen, Jiaojiao Zhao, Yunhwa Zhang, Teng Long, Pengwan Zhang.

Besides dutch and east Asian culture Devanshu Arya and Deepak Gupta were a gateway back to south Asia for me. I am thankful to both of them for providing me their suggestions from the east Asian perspective. Moreover, I am thankful to them for inviting me for Indian food at their homes.

I am grateful to other members in the Vislab Pascal Mettes, David Zhang, Adeel Pervaiz, Fida Thoker, Riaan Zoetmulder, Efstratios Gavves, Dennis Koelma, Mina Ghadimiatigh, Tejaswi Kasarla, Sarah Rastegar, Mehmet Altinkaya, Melika Ayoughi, Noureldien Hussein and others for their companionship and guidance at every stage of the PhD.

I am thankful to people from Amlab, Karen Ullrich, Maximilian Ilse, Patrick Forre, Tim Bakker, Marco Federici, Daniel Worrel, Bas Veeling and Jakub Tomczak. Whether through their technical discussions during the seminars or general discussions regarding

diversity in AI, inclusive environment, global warming and many more during coffee and lunch breaks, I got to learn a lot.

I am also grateful to my current lab members Marcel Worring Zeno Geradts, Stevan Rudinac, Nanne van Noord, Yen-Chia Hsu, Teng Long, Inske Groenen, Floris Gisolf, Sarah Ibrahimi, Jia-Hong Huang, Andrea Macarulla, Eleni Konstantina, Jiayi Shen, Ivona Najdenkoska, Tom van Sonsbeek, Meike Kombrink, Ujjwal Sharma, Thanos Efthimiou, Wangyuan Ding, Tim Alpherts, Shuai Wang, Carlo Bretti. Many thanks to Marcel for letting me be the part of his lab and facilitating me with every type of facility I need. I am also grateful to Nanne for providing me with the opportunity of Postdoc and guiding me throughout. Many thanks to Teng for his help until now, I have learned a lot from him through our collaboration. Thanks a lot to Tom for translating my thesis summary. I am grateful to Dennis Koelma for all the technical support that he provided us. Both the PhD and Postdoc would not have been possible without his support.

I would also like to thank those friends who were not directly the part of my PhD but were there for me whenever I needed them, Jauwairia Nasir and Gehan Fatima. Last but not least I am thankful to my family, my parents for their prayers and unconditional support and understanding. I am thankful to my sister Khansa and my brother Umar for their suggestions and support throughout the journey.

BIBLIOGRAPHY

[1] swov.nl.

[2] theigc.org.

[3] A. A. Alsanabani, M. A. Ahmed, and A. M. Al Smadi. Vehicle counting using detecting-tracking combinations: A comparative analysis. In *2020 The 4th International Conference on Video and Image Processing*, pages 48–54, 2020.

[4] E. Abbasnejad, D. Teney, A. Parvaneh, J. Shi, and A. v. d. Hengel. Counterfactual vision and language learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10044–10054, 2020.

[5] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 819–826, 2013.

[6] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015.

[7] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[8] L. Anne Hendricks, R. Hu, T. Darrell, and Z. Akata. Grounding visual explanations. In *ECCV*, 2018.

[9] A. Azulay and Y. Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018.

[10] E. Becic, N. Zych, and J. Ivarsson. Vehicle automation report hwy18mh010. *NATIONAL TRANSPORTATION SAFETY BOARD*, 2019.

[11] P. Benz, C. Zhang, A. Karjauv, and I. S. Kweon. Revisiting batch normalization for improving corruption robustness. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 494–503, 2021.

[12] A. Bhattad, M. J. Chong, K. Liang, B. Li, and D. A. Forsyth. Unrestricted adversarial examples via semantic manipulation. *arXiv preprint arXiv:1904.06347*, 2019.

[13] S. Bhojanapalli, A. Chakrabarti, D. Glasner, D. Li, T. Unterthiner, and A. Veit. Understanding robustness of transformers for image classification. *arXiv preprint arXiv:2103.14586*, 2021.

[14] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.

[15] brgfx. www.freepik.com.

[16] K. Browne and B. Swift. Semantics and explanation: why counterfactual explanations produce adversarial examples in deep neural networks, 2020.

[17] J. Bruna and S. Mallat. Invariant scattering convolution networks. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1872–1886, 2013.

[18] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[19] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *SP*. IEEE, 2017.

[20] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks, 2017.

[21] N. Carlini and D. Wagner. Audio adversarial examples: Targeted attacks on speech-to-text, 2018.

[22] C. Chen, O. Li, D. Tao, A. Barnett, C. Rudin, and J. K. Su. This looks like that: deep learning for interpretable image recognition. *Advances in neural information processing systems*, 32, 2019.

[23] H. Chen, A. Gallagher, and B. Girod. What's in a name: First names as facial attributes. In *Proc. CVPR*, 2013.

[24] A. Coates, A. Ng, and H. Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[25] T. Cohen and M. Welling. Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR, 2016.

[26] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.

[27] J. Deng, W. Dong, R. Socher, L. 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, 2009.

[28] T. DeVries and G. W. Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[29] S. Dodge and L. Karam. Understanding how image quality affects deep neural networks. In *2016 eighth international conference on quality of multimedia experience (QoMEX)*, pages 1–6. IEEE, 2016.

[30] S. Dodge and L. Karam. Quality resilient deep neural networks. *arXiv preprint arXiv:1703.08119*, 2017.

[31] S. Dodge and L. Karam. A study and comparison of human and deep learning recognition performance under visual distortions. In *2017 26th international conference on computer communication and networks (ICCCN)*, pages 1–7. IEEE, 2017.

[32] Y. Dong, Q.-A. Fu, X. Yang, T. Pang, H. Su, Z. Xiao, and J. Zhu. Benchmarking adversarial robustness on image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 321–331, 2020.

[33] Y. Dong, H. Su, J. Zhu, and B. Zhang. Improving interpretability of deep neural networks with semantic information. In *CVPR*, 2017.

[34] J. Donnelly, A. J. Barnett, and C. Chen. Deformable protopnet: An interpretable image classifier using deformable prototypes. *arXiv preprint arXiv:2111.15000*, 2021.

[35] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[36] M. Du, N. Liu, and X. Hu. Techniques for interpretable machine learning. *arXiv*, 2018.

[37] L. Edwards and M. Veale. Slave to the algorithm: Why a right to an explanation is probably not the remedy you are looking for. *Duke L. & Tech. Rev.*, 16:18, 2017.

[38] L. Engstrom, B. Tran, D. Tsipras, L. Schmidt, and A. Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning*, pages 1802–1811. PMLR, 2019.

[39] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *2009 IEEE conference on computer vision and pattern recognition*, pages 1778–1785. IEEE, 2009.

[40] A. Fawzi and P. Frossard. Manitest: Are classifiers really invariant? *arXiv preprint arXiv:1507.06535*, 2015.

[41] X. Y. Felix, R. Ji, M.-H. Tsai, G. Ye, and S.-F. Chang. Weak attributes for large-scale image retrieval. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2949–2956. IEEE, 2012.

[42] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2009.

[43] R. S. Feris, C. Lampert, and D. Parikh. *Visual Attributes*. Springer, 2017.

[44] V. Ferrari and A. Zisserman. Learning visual attributes. *Advances in neural information processing systems*, 20:433–440, 2007.

[45] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *arXiv*, 2017.

[46] N. Ford, J. Gilmer, N. Carlini, and D. Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019.

[47] L. A. Gatys, A. S. Ecker, and M. Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[48] R. Geirhos, D. H. Janssen, H. H. Schütt, J. Rauber, M. Bethge, and F. A. Wichmann. Comparing deep neural networks against humans: object recognition when the signal gets weaker. *arXiv preprint arXiv:1706.06969*, 2017.

[49] T. Gokhale, R. Anirudh, B. Kailkhura, J. J. Thiagarajan, C. Baral, and Y. Yang. Attribute-guided adversarial training for robustness to natural perturbations. *arXiv preprint arXiv:2012.01806*, 2020.

[50] I. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[51] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[52] Y. Goyal, Z. Wu, J. Ernst, D. Batra, D. Parikh, and S. Lee. Counterfactual visual explanations. In *International Conference on Machine Learning*, pages 2376–2384. PMLR, 2019.

[53] S. Gulshad and A. Smeulders. Explaining with counter visual attributes and examples. In *Proceedings of the 2020 International Conference on Multimedia Retrieval*, pages 35–43, 2020.

[54] S. Gulshad and A. Smeulders. Counterfactual attribute-based visual explanations for classification. *International Journal of Multimedia Information Retrieval*, 10(2):127–140, 2021.

[55] S. Gulshad and A. Smeulders. Natural perturbed training for general robustness of neural network classifiers. *arXiv preprint arXiv:2103.11372*, 2021.

[56] S. Gulshad, I. Sosnovik, and A. Smeulders. Wiggling weights to improve the robustness of classifiers. *arXiv preprint arXiv:2111.09779*, 2021.

[57] D. Gunning, M. Stefik, J. Choi, T. Miller, S. Stumpf, and G.-Z. Yang. Xai—explainable artificial intelligence. *Science Robotics*, 4(37), 2019.

[58] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[59] L. A. Hendricks, Z. Akata, M. Rohrbach, J. Donahue, B. Schiele, and T. Darrell. Generating visual explanations. In *ECCV*. Springer, 2016.

[60] L. A. Hendricks, R. Hu, T. Darrell, and Z. Akata. Generating counterfactual explanations with natural language. In *ICML Workshop on Human Interpretability in Machine Learning*, pages 95–98, 2018.

[61] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. *arXiv preprint arXiv:2006.16241*, 2020.

[62] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. 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.

[63] D. Hendrycks and T. Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *arXiv preprint arXiv:1903.12261*, 2019.

[64] D. Hendrycks, K. Lee, and M. Mazeika. Using pre-training can improve model robustness and uncertainty. In *International Conference on Machine Learning*, pages 2712–2721. PMLR, 2019.

[65] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

[66] G. F. Hinton. A parallel computation that assigns canonical object-based frames of reference. In *Proceedings of the 7th international joint conference on Artificial intelligence-Volume 2*, pages 683–685, 1981.

[67] H. Hosseini and R. Poovendran. Semantic adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1614–1619, 2018.

[68] C.-Y. Hsieh, C.-K. Yeh, X. Liu, P. Ravikumar, S. Kim, S. Kumar, and C.-J. Hsieh. Evaluations and methods for explanation through robustness analysis, 2020.

[69] A. Ignatiev, N. Narodytska, and J. Marques-Silva. On relating explanations and adversarial examples. In *Advances in Neural Information Processing Systems*, pages 15883–15893, 2019.

[70] J.-H. Jacobsen, J. Van Gemert, Z. Lou, and A. W. Smeulders. Structured receptive fields in cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2610–2619, 2016.

[71] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

[72] L. Jiang, S. Liu, and C. Chen. Recent research advances on interactive machine learning. *Journal of Visualization*, 2018.

[73] A. Kanazawa, A. Sharma, and D. Jacobs. Locally scale-invariant convolutional neural networks. *arXiv preprint arXiv:1412.5104*, 2014.

[74] C. Kanbak, S.-M. Moosavi-Dezfooli, and P. Frossard. Geometric robustness of deep networks: analysis and improvement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4441–4449, 2018.

[75] A. Kanehira and T. Harada. Learning to explain with complemental examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8603–8611, 2019.

[76] S. Karahan, M. K. Yildirum, K. Kirtac, F. S. Rende, G. Butun, and H. K. Ekenel. How image degradations affect deep cnn-based face recognition? In *2016 international conference of the biometrics special interest group (BIOSIG)*, pages 1–5. IEEE, 2016.

[77] A. Kovashka, D. Parikh, and K. Grauman. Whittlesearch: Interactive image search with relative attribute feedback. *International Journal of Computer Vision*, 115(2):185–210, 2015.

[78] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

[79] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, 2017.

[80] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[81] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[82] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *ICLR workshop*, 2017.

[83] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*. IEEE, 2009.

[84] D. Laptev, N. Savinov, J. M. Buhmann, and M. Pollefeys. Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 289–297, 2016.

[85] A. Laugros, A. Caplier, and M. Ospici. Are adversarial robustness and common perturbation robustness independant attributes? In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[86] A. LAUGROS, A. Caplier, and M. Ospici. Increasing the coverage and balance of robustness benchmarks by using non-overlapping corruptions, 2021.

[87] S. Liu, B. Kailkhura, D. Loveland, and Y. Han. Generative counterfactual introspection for explainable deep learning. *arXiv preprint arXiv:1907.03077*, 2019.

[88] Z. Liu, P. Luo, X. Wang, and X. Tang. Large-scale celebfaces attributes (celeba) dataset. *Retrieved August*, 15(2018):11, 2018.

[89] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[90] O. Loyola-González. Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view. *IEEE Access*, 7:154096–154113, 2019.

[91] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

[92] D. Mahajan, S. Sellamanickam, and V. Nair. A joint learning framework for attribute models and object descriptions. In *2011 International Conference on Computer Vision*, pages 1227–1234. IEEE, 2011.

[93] K. Mahmood, R. Mahmood, and M. Van Dijk. On the robustness of vision transformers to adversarial examples. *arXiv preprint arXiv:2104.02610*, 2021.

[94] X. Mao, G. Qi, Y. Chen, X. Li, R. Duan, S. Ye, Y. He, and H. Xue. Towards robust vision transformer. *arXiv preprint arXiv:2105.07926*, 2021.

[95] J. S. J. Melrose, B. Madahar, M. Aktaş, N. Martinel, J. de Marchi, E. Solberg, D. S. Lange, G. O. Tanik, F. Kurth, and L. Luotsinen. Robustness of artificial intelligence for hybrid warfare.

[96] T. Miyato, S.-i. Maeda, M. Koyama, and S. 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.

[97] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

[98] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *EuroS&P*. IEEE, 2016.

[99] D. H. Park, L. A. Hendricks, Z. Akata, B. Schiele, T. Darrell, and M. Rohrbach. Multimodal explanations: Justifying decisions and pointing to the evidence. In *CVPR*, 2018.

[100] G. Patterson and J. Hays. Coco attributes: Attributes for people, animals, and objects. *European Conference on Computer Vision*, 2016.

[101] P. J. Phillips, C. A. Hahn, P. C. Fontana, D. A. Broniatowski, and M. A. Przybocki. Four principles of explainable artificial intelligence. *Gaithersburg, Maryland*, 2020.

[102] B. Recht, R. Roelofs, L. Schmidt, and V. Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018.

[103] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 49–58, 2016.

[104] M. T. Ribeiro, S. Singh, and C. Guestrin. Why should i trust you?: Explaining the predictions of any classifier. In *ACM SIGKDD*, 2016.

[105] A. Robey, H. Hassani, and G. J. Pappas. Model-based robust deep learning. *arXiv preprint arXiv:2005.10247*, 2020.

[106] E. Rusak, L. Schott, R. Zimmermann, J. Bitterwolf, O. Bringmann, M. Bethge, and W. Brendel. Increasing the robustness of dnns against image corruptions by playing the game of noise. *arXiv preprint arXiv:2001.06057*, 2020.

[107] E. Rusak, L. Schott, R. S. Zimmermann, J. Bitterwolf, O. Bringmann, M. Bethge, and W. Brendel. A simple way to make neural networks robust against diverse image corruptions. In *European Conference on Computer Vision*, pages 53–69. Springer, 2020.

[108] D. Rymarczyk, Ł. Struski, J. Tabor, and B. Zieliński. Protopshare: Prototypical parts sharing for similarity discovery in interpretable image classification. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1420–1430, 2021.

[109] S. Schneider, E. Rusak, L. Eck, O. Bringmann, W. Brendel, and M. Bethge. Improving robustness against common corruptions by covariate shift adaptation. *Advances in Neural Information Processing Systems*, 33, 2020.

[110] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.

[111] R. Shao, Z. Shi, J. Yi, P.-Y. Chen, and C.-J. Hsieh. On the adversarial robustness of visual transformers. *arXiv preprint arXiv:2103.15670*, 2021.

[112] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *ICML*, 2017.

[113] B. Siddiquie, R. S. Feris, and L. S. Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR 2011*, pages 801–808. IEEE, 2011.

[114] P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer, 1998.

[115] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv*, 2013.

[116] C. Song, K. He, J. Lin, L. Wang, and J. E. Hopcroft. Robust local features for improving the generalization of adversarial training. *arXiv preprint arXiv:1909.10147*, 2019.

[117] I. Sosnovik, A. Moskalev, and A. Smeulders. Disco: accurate discrete scale convolutions. *arXiv preprint arXiv:2106.02733*, 2021.

[118] I. Sosnovik, A. Moskalev, and A. W. Smeulders. Scale equivariance improves siamese tracking. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2765–2774, 2021.

[119] I. Sosnovik, M. Szmaja, and A. Smeulders. Scale-equivariant steerable networks. *arXiv preprint arXiv:1910.11093*, 2019.

[120] D. Su, H. Zhang, H. Chen, J. Yi, P.-Y. Chen, and Y. Gao. Is robustness the cost of accuracy?–a comprehensive study on the robustness of 18 deep image classification models. In *ECCV*, 2018.

[121] J. Su, D. V. Vargas, and K. Sakurai. One pixel attack for fooling deep neural networks. *TEVC*, 2019.

[122] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *ICLR*, 2013.

[123] Z. Tang, Y. Gao, Y. Zhu, Z. Zhang, M. Li, and D. Metaxas. Selfnorm and crossnorm for out-of-distribution robustness. *arXiv preprint arXiv:2102.02811*, 2021.

[124] R. Tao, A. W. Smeulders, and S.-F. Chang. Attributes and categories for generic instance search from one example. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 177–186, 2015.

[125] M. N. Teli and S. Oh. Resilience of autonomous vehicle object category detection to universal adversarial perturbations. In *2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS)*, pages 1–6. IEEE, 2021.

[126] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. *stat*, 1050, 2018.

[127] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[128] vectorpouch. www.freepik.com.

[129] S. Wachter, B. Mittelstadt, and C. Russell. Counterfactual explanations without opening the black box: Automated decisions and the gdpr. *Harv. JL & Tech.*, 31:841, 2017.

[130] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[131] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes and visual saliency. In *2009 IEEE 12th International Conference on Computer Vision*, pages 537–544. IEEE, 2009.

[132] H. Wang, Y. Zhu, H. Adam, A. Yuille, and L.-C. Chen. Max-deeplab: End-to-end panoptic segmentation with mask transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5463–5474, 2021.

[133] Y. Wang, Z. Xu, X. Wang, C. Shen, B. Cheng, H. Shen, and H. Xia. End-to-end video instance segmentation with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8741–8750, 2021.

[134] M. Weiler, F. A. Hamprecht, and M. Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 849–858, 2018.

[135] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[136] E. Wong and J. Z. Kolter. Learning perturbation sets for robust machine learning. *arXiv preprint arXiv:2007.08450*, 2020.

[137] E. Wong, L. Rice, and J. Z. Kolter. Fast is better than free: Revisiting adversarial training. *arXiv preprint arXiv:2001.03994*, 2020.

[138] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[139] D. E. Worrall and M. Welling. Deep scale-spaces: Equivariance over scale. *arXiv preprint arXiv:1905.11697*, 2019.

[140] 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.

[141] W. Xu, Y. Xian, J. Wang, B. Schiele, and Z. Akata. Attribute prototype network for zero-shot learning. *arXiv preprint arXiv:2008.08290*, 2020.

[142] J. Yang, C. Li, P. Zhang, X. Dai, B. Xiao, L. Yuan, and J. Gao. Focal attention for long-range interactions in vision transformers. *Advances in Neural Information Processing Systems*, 34, 2021.

[143] D. Yin, R. G. Lopes, J. Shlens, E. D. Cubuk, and J. Gilmer. A fourier perspective on model robustness in computer vision. *arXiv preprint arXiv:1906.08988*, 2019.

[144] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013.

[145] L. Yuan, Y. Chen, T. Wang, W. Yu, Y. Shi, Z. Jiang, F. E. Tay, J. Feng, and S. Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

[146] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6023–6032, 2019.

[147] H. Zhang, H. Chen, Z. Song, D. Boning, I. S. Dhillon, and C.-J. Hsieh. The limitations of adversarial training and the blind-spot attack. *arXiv preprint arXiv:1901.04684*, 2019.

[148] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[149] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, pages 7472–7482. PMLR, 2019.

[150] T. Zhang and Z. Zhu. Interpreting adversarially trained convolutional neural networks. *arXiv*, 2019.

[151] B. Zhao, Y. Fu, R. Liang, J. Wu, Y. Wang, and Y. Wang. A large-scale attribute dataset for zero-shot learning. *arXiv*, 2018.

[152] B. Zhao, Y. Fu, R. Liang, J. Wu, Y. Wang, and Y. Wang. A large-scale attribute dataset for zero-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[153] L. M. Zintgraf, T. S. Cohen, T. Adel, and M. Welling. Visualizing deep neural network decisions: Prediction difference analysis. *ICLR*, 2017.