TECHNISCHE UNIVERSITÄT
CHEMNITZ

Learning Shape Based Features for Robust Neural Representation

Master Thesis

Ranjitha Subramaniam
Matriculation Number: 516518

Submitted in partial fulfilment for the award of the degree of

M.Sc. in Automotive Software Engineering

Department of Computer Science
Professorship for Artificial Intelligence

Supervisors:
Dr. habil. Julien Vitay , TU Chemnitz
Dr.Jan Hendrik Metzen, Bosch Center for Artificial Intelligence

June 23rd, 2020

Acknowledgements

I would first like to thank my supervisors, Dr. Jan Hendrik Metzen and Mr. Chaithanya Kumar Mummadi, Bosch Center for Artificial Intelligence (BCAI), for their immense support and guidance throughout the work. With regular technical discussions and feedback, they assisted me to proceed in the right direction.

I wish to express my gratitude to my professor, Dr. habil. Julien Vitay, TU Chemnitz, for his advice and suggestions throughout the thesis. Particularly, I would like to acknowledge his prompt responses to my ceaseless queries.

I am grateful to all direct and indirect contributions of my colleagues at Robert Bosch for the thesis work. Finally, a special thanks to my friends and family for their encouragement and unconditional support at all times.

June 23, 2020 **Ranjitha Subramaniam**

Abstract

Convolutional Neural Networks (CNNs) have gained tremendous significance over the years with state-of-the-art results in many computer vision tasks like object recognition, object detection, semantic segmentation, etc. Such high performance of CNNs is commonly attributed to the fact that they learn increasingly complex features while traversing deeper in their layers and this behavior is analogous to how humans perceive objects. Nevertheless, recent studies revealed that there exist considerable differences between human visual perception and the perception of objects by CNNs. One such substantial distinction is that humans predominantly rely on robust shape features to recognize objects while CNNs are highly biased towards local texture cues for object recognition [1, 2].

The perceptional differences between CNNs and humans can be reduced by improving the shape bias of CNNs. Recent work from Geirhos et al. [1] showed that the augmentation of natural images using various styles from paintings makes their texture cues unpredictable and enforces the networks to learn more robust features. A CNN trained on such stylized images exhibits improved shape bias than a standard network trained on natural images. Besides the enhanced shape bias, such a network also demonstrates improved robustness against common image corruptions such as noise, blur, etc. The improved shape bias of the network is hypothesized to be the reason behind its high corruption robustness.

With the objective to improve shape bias of CNNs, a technique, which employs edge maps with explicit shape details, is introduced in this thesis work. Moreover, the possible texture bias of the network is reduced by a technique called style randomization, which randomizes the statistics of activation maps in feature space. On evaluation, the proposed network shows higher shape bias. However, this shape biased network displays poor performance on image corruptions and its results are no better than a standard texture biased CNN. Hence, a systematic study is carried out to analyze the different characteristics in an image that could influence the corruption robustness. These characteristics include the existence of natural image properties, explicit shape details from edge maps and the stylized texture details. While stylization and certain preserved statistics of natural images play a role in improving the corruption robustness, no clear correlation is observed between the shape bias of a CNN and its corruption robustness. This study reveals that the strong data augmentation, which resulted from the stylization of natural images, helped in improving the corruption robustness of stylized networks while their improvement in shape bias emerged only as a byproduct.

A further study is conducted to understand the adaptability of a network pretrained on natural images to data from different distributions. It is observed that the network shows improved performance on the target data while finetuning only its affine parameters of normalization layers. This indicates that a network trained using natural images also encodes robust representations but these representations are not leveraged in its affine layers.

Contents

List of Figures i
List of Tables iii
List of Abbreviations iv

1 Introduction **1**
1.1 Motivation 1
1.1.1 Texture vs Shape Bias 2
1.1.2 Robustness on Corruptions 3
1.2 Problem Statement 3
1.3 Proposed Solution 4
1.4 Thesis Structure 5

2 Background **7**
2.1 Convolutional Neural Networks 7
2.1.1 Normalization Techniques 8
2.2 Robustness 12
2.2.1 Understanding Shape Details 12
2.2.2 Robustness to Common Corruptions 12
2.2.3 Robustness against Adversarial Examples 13

3 Related Work **14**
3.1 Studying the Behavior of CNNs 14
3.2 Enhancing Robust Representations of CNNs 17
3.3 Robustness against Common Corruptions 24
3.4 Edge Detection 26

4 Learning Shape Representations **28**
4.1 Dataset - ImageNet20 28
4.2 Learning Shape Representations using Edge Maps 29
4.3 Style Randomization 32
4.4 Style-based Data Augmentations 32

5 Evaluation Strategies **36**
5.1 Shuffled Image Patches 36
5.2 Texture-Shape Cue Conflict Images 37
5.3 ImageNet-C Corruptions 38

6 Experiments and Results **41**
6.1 Experimental Settings 41
6.2 Results and Analysis 43
6.2.1 *IN* vs *EdgeCNN* 43
6.2.2 Variants of *EdgeCNN* 44
6.2.3 Evaluation of the Shape based Network (*E*) 46
6.2.4 Effects of Style Randomization 48
6.2.5 Evaluation of *IN, SIN, E, SE* & *E-SIN* with Style Randomization . 50
6.2.6 Evaluation of *IN, SIN* & *E* on ImageNet-C distortions 53
6.2.7 Influence of Shape Bias on Common Corruptions 55
6.2.8 On the Adaptability of Learned Representations 60

7 Conclusion and outlook **63**
7.1 Conclusion 63
7.1.1 Enhancing the Shape Bias of CNNs 63
7.1.2 Evaluation of Shape Bias 64
7.1.3 Shape Bias doesn't Improve Corruption Robustness 64
7.1.4 Adaptability of Learned Representations 64
7.2 Outlook 65

Bibliography **66**

List of Figures

1.1 Predictions of a pretrained CNN: (a) elephant texture (b) cat content (c) cue conflict image with elephant texture and cat shape (source: R. Geirhos, 2018 [1]). 2
1.2 Pictorial demonstration of the activities involved in this thesis work. 5

2.1 Different Normalization techniques [3]. 8
2.2 AdaIN style transfer network [4]. 10
2.3 An example of Adaptive Instance Normalization (AdaIN) style transfer with varying degrees of stylization [4]. 11
2.4 Weight Standardization [5]. 11
2.5 Adversarial example generation [6]. 13

3.1 Example images for each experimental setting considered in [2]. 15
3.2 Different images accounted for the experiments of [1] along with their prediction accuracies by various Convolutional Neural Network (CNN) architectures as well as humans. 16
3.3 Illustration of image stylization [1]. 18
3.4 Defective convolutional layers [7]. 19
3.5 Style adversarial learning framework [8]. 20
3.6 Salience maps from standard, underfitting and AT-CNNs on original, saturated and stylized images [9]. 23
3.7 Visual comparison of different data augmentation techniques [10]. 24
3.8 An example of AugMix operation [10]. 25
3.9 Edge outputs of varying details with Rich Convolutional Features (RCF) edge detection [11]. 26
3.10 Canny vs. RCF edges. 27

4.1 ImageNet20 dataset classes. 29
4.2 Different variants of edge maps considered in the thesis work. 30
4.3 Different image variants considered for evaluating corruption robustness. . . 33
4.4 Superposition images at different values of α. 35

5.1 An elephant image along with its shuffled patches. 37
5.2 Texture-Shape Cue Conflict images. 38
5.3 Different corruptions included in ImageNet-C dataset [12]. 39
5.4 An image distorted with Impulse Noise at different severity levels [12]. 40

6.1 Evaluation results of IN and EdgeCNN on corresponding shuffled image patches. Lower accuracy indicates higher shape bias. 43
6.2 Evaluation results of different *EdgeCNN* variants on corresponding shuffled image patches. Lower accuracy indicates higher shape bias. 44
6.3 shuffled patch results for different EdgeCNN variants on IN dataset. Lower accuracy indicates higher shape bias. 45
6.4 Evaluation results of *IN, SIN* and *E* on shuffled image patches of IN dataset. Lower accuracy indicates higher shape bias. 46
6.5 Shuffled patch results of *IN, SIN* and *E* on the correctly classified images of IN dataset. Lower accuracy indicates higher shape bias. 47
6.6 Shuffled patch accuracies of the networks *IN, SIN, E, SE,* and *E-SIN* with style randomization on the entire validation dataset of IN. Lower accuracy indicates higher shape bias. 50
6.7 Shuffled patches accuracies of the networks *IN, SIN, E, SE,* and *E-SIN* with style randomization on the correctly classified validation images of IN dataset. Lower accuracy indicates higher shape bias. 51
6.8 Performance of *IN, SIN* & *E* on the validation distortions of ImageNet-C at different severity levels. 54
6.9 Performance of various networks on the validation distortions of ImageNet-C at different severity levels. 57
6.10 Mean corruption accuracy of superposition network variants (SE+IN) trained using images with different α values. 59
6.11 Performance of standard network on ImageNet-C corruptions along with its finetuned variants on respective distortions. 61

List of Tables

6.1 Comparison of texture-shape cue conflict results between *IN, SIN* and *E*. . . 48
6.2 Comparison of different feature space style augmentation methods on 4×4 shuffled image patches. Lower accuracy indicates higher shape bias. 49
6.3 Comparison between different feature space style augmentation methods for the shape based results of cue conflict images. 49
6.4 Comparison between different networks *IN, SIN, E, SE,* and *E-SIN* with style randomization for the shape based classification of 400 cue conflict images. . 52
6.5 Comparison of texture and shape results between different networks *IN, SIN, E, SE,* and *E-SIN* with style randomization on 100 cue conflict images. . . . 52
6.6 Comparison of the validation accuracies of *IN, SIN* & *E* on different categories of ImageNet-C distortions. 53
6.7 Comparison of the validation accuracies of various networks on different categories of ImageNet-C distortions. 56
6.8 Analysis of different networks using their input compositions, texture/shape accuracy on cue conflict images, and Mean corruption accuracy. 58
6.9 Corruption and SIN accuracies along with the cue conflict results for the networks with and without finetuning of their affine parameters on respective datasets. 62

List of Abbreviations

AdaIN Adaptive Instance Normalization

AT-CNN Adversarially Trained CNN

BN Batch Normalization

BoF Bag-of-Feature

CNN Convolutional Neural Network

E Edges of ImageNet

FGSM Fast Gradient Sign Method

GN Group Normalization

GPU Graphics Processing Unit

I-SE Intra-Stylized Edges

I-SIN Intra-Stylized ImageNet

IN ImageNet

IN Instance Normalization

LN Layer Normalization

mCA mean Corruption Accuracy

NLP Natural Language Processing

PAR Patch-wise Adversarial Regularization

RCF Rich Convolutional Features

ReLU Rectified Linear Unit

RNN Recurrent Neural Network

SagNet Style-Agnostic Network

SE Stylized Edges

SGD Stochastic Gradient Descent

SIN Stylized ImageNet

tanH Hyperbolic tangent

WS Weight Standardization

1 Introduction

In recent years, deep learning has gained momentum within the field of artificial intelligence and showed significant progress in topics like Natural Language Processing (NLP), computer vision, etc., This breakthrough of deep learning is due to two important factors: the availability of large amount of labeled data and the processing power of modern-day Graphics Processing Units (GPUs). Further, the introduction of CNNs caught attention in the research world as they achieved state-of-the-art results in many computer vision tasks. Recent CNN architectures also attained human-level performance on tasks like image classification, object detection, semantic segmentation, etc. As a result of their high performance, CNNs continue to gain their place in many real-world applications including safety-critical domains like autonomous driving. However, recent studies revealed that such powerful CNNs also possess a risk factor. These complex networks can get fooled when slight modifications are introduced on the input images, though such changes are visually imperceptible to humans. The discovery of the existence of aforesaid examples, known as adversarial images, has brought the study of CNNs' robustness into focus. Unless the deep neural networks are robust enough, they cannot be utilized in any of the safety-critical domains. Recent works compared the visual perception of humans with CNNs and revealed a surprising fact that there exist huge behavioral differences between them. It is observed that the CNNs, which are once thought to learn increasingly complex features, indeed rely on simple cues such as texture details for their classification and they do not focus on robust features such as shape which humans actually rely on.

1.1 Motivation

Human visual perception has been studied over decades and there exist plenty of contributions which analyze the behavioral pattern of humans in recognizing objects. One of the interesting findings is that humans rely on more complex and robust features, particularly the shape cues, for recognizing objects [13]. Convolutional Neural Networks (CNNs), which achieved human-level performance in object recognition tasks, are also anticipated to focus on such robust features for object recognition. Considering the presumed fact that CNNs emulate human visual perception by learning complex features in their deeper layers, CNNs are believed to be the possible substitute of humans in scenarios like autonomous driving. However, recent works [1, 2, 14] contradict the earlier made behavioral assumptions of CNNs.

1.1.1 Texture vs Shape Bias

Baker et al. [2] conducted a detailed study to understand if the object recognition pattern of CNNs matches the visual perception of humans and found that CNNs rely on local surface statistical cues such as textures for classifying objects while humans focus on shape details. When evaluated on images whose texture details are modified without any changes to their global shape, CNNs failed to recognize the objects owing to their texture bias. However, humans had no difficulty in such scenarios as the object shape is preserved. Conversely, when the global shape of the objects is altered, humans can no more recognize them correctly while CNNs could still identify such objects. Besides, Geirhos et al. [1] made an extensive evaluation of the texture vs shape bias between humans and CNNs by introducing a special kind of images called texture-shape cue conflict images. As the name indicates, texture-shape cue conflict images possess shape and texture features from two different image categories. When such images are considered for evaluation, the results could either be shape biased or texture biased, depending on the internal representations of the network. Supporting the recent claims, results of a CNN pretrained on natural images are texture biased while humans categorized them according to their shape labels.

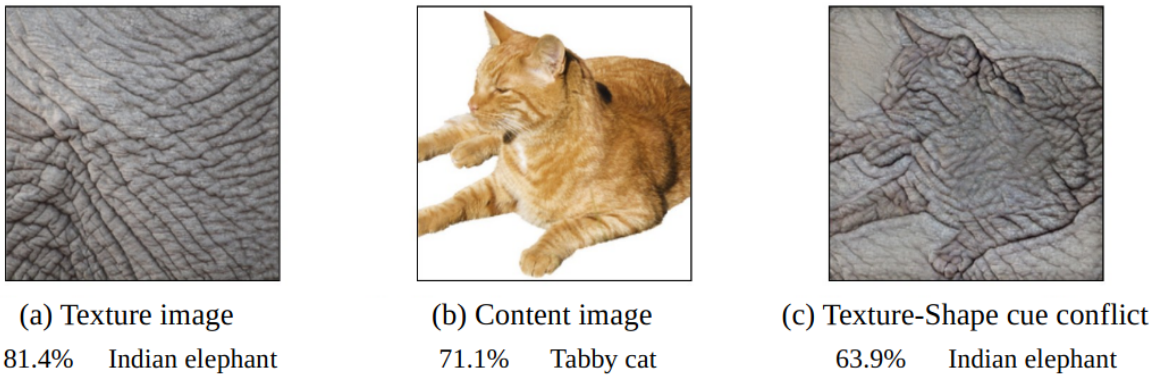

Figure 1.1: Predictions of a pretrained CNN: (a) elephant texture (b) cat content (c) cue conflict image with elephant texture and cat shape (source: R. Geirhos, 2018 [1]).

Figure 1.1(c) represents a texture-shape cue conflict image formulated from the two images, (a) and (b), on its left. Image (a) represents the texture of an elephant while image (b) represents a cat structure. The texture-shape cue conflict image (c) is generated through the style transfer of elephant texture(a) to the cat image(b). While we humans would predict this cue conflict image to be a cat, it can be seen in Figure 1.1(c) that a pretrained CNN predicted it to be an elephant with almost 64% confidence. Hence, the prime motivation behind this thesis work is to close this existing gap between humans and CNNs by enforcing the CNNs to rely on robust shape details for decision making.

1.1.2 Robustness on Corruptions

Natural images are generally subjected to many sorts of commonly occuring distortions such as noise, weather changes, etc. Humans are essentially insusceptible to such image corruptions and should be able to correctly recognize the corrupted images. On the other hand, recent studies [12, 14] revealed that the deep neural networks trained on natural images could not generalize well to those images that are subjected to common corruptions. Hendrycks et al. [12] introduced a new benchmark dataset called ImageNet-C that incorporates different corruptions from categories like noise, blur, weather effects and digital image transformations. During the evaluation of a pretrained CNN on these image corruptions, a significant drop in its accuracy is observed. In addition, Geirhos et al. [14] showed that humans withstand most of the distortions and they perform far better than CNNs. This illustrates another behavioral difference between humans and CNNs. One hypothesis for this reduced robustness of CNNs is attributed to their over-reliance on local features such as texture which aid them to generalize well within the input data distribution but doesn't help with out-of-distribution images. It has been further hypothesized that the networks that rely on robust shape features show improved corruption robustness. For example, Geirhos et al. [1] introduced a new dataset by stylizing natural images using the styles from paintings. As texture cues in such images are no more meaningful, a network trained on such stylized images exhibits stronger shape bias. Further, such network also shows improved corruption robustness. The enhanced shape bias of the network is hypothesized to be the reason behind its improved corruption robustness. However, other than the enhanced shape bias, the robustness of CNNs against common corruptions can be improved using different methods like self-training with more training data, using stronger data augmentation methods, etc,. Further, the stylized images could also be interpreted as a kind of data augmentation. The results on stylized images motivated us to conduct a detailed study on the actual relationship between the shape bias of a network and its performance on different distortions.

1.2 Problem Statement

It has been discussed that there exist potential differences between the visual perception of humans and the recognition of objects by deep neural networks. In short, humans interpret objects through their shapes while CNNs exhibit high texture bias. Additionally, humans could recognize objects in the distorted images whereas CNNs suffer a performance drop on such images. Further, the lack of robust representations like shape in the CNNs is hypothesized to be the reason behind their reduced corruption robustness. Considering these facts, this thesis work addresses the following questions:

1. Is there a simple way to enhance the robust representations of CNNs so that they focus on object structures rather than the local surface statistical regularities like texture?

2. Does the proposed approach show improved shape bias on CNNs when compared to the existing networks?

3. When a network possesses improved shape bias, whether this automatically aids in improving the network's robustness against common corruptions?
4. Besides shape features, what properties of an image could possibly contribute to improved corruption robustness?
5. If not shape bias, are there other representations that would help the pretrained networks in adapting to different distributions without modifying the already learned feature extractors?

1.3 Proposed Solution

This thesis work introduces an approach to improve the shape based representations of a CNN by using explicit shape details from edge maps. Edges are fast gradient changes within the images and they represent object boundaries. Therefore, they preserve the complete shape information of objects and are devoid of other object cues such as texture, color, brightness etc. Hence, a network trained using such edge maps could focus only on the edge contours to differentiate between the objects. Moreover, as the distribution of edge maps differs greatly than that of original images, in order to generalize the edge map trained networks on natural images, they have to be further finetuned for considerable iterations on natural images. However, this finetuning may allow the network to easily encode the texture details. Hence, in order to reduce this texture bias, a new method called style randomization is proposed in this work. This technique randomizes the style information in feature space thereby forcefully distracting the network from focusing on local texture details. The shape based network proposed in this thesis work is shown to exhibit stronger shape bias than existing approaches like stylized ImageNet [1] with the help of the following evaluation strategies:

- Testing the networks on images where the global shape details are destroyed while only their local texture details are preserved. Such images, termed as patch shuffled images, are formulated by splitting the original images into a number of equal-sized patches and then randomly shuffling those patches before rejoining them. Figure 5.1 depicts the shuffled patches of an elephant image. A network with high shape bias is expected to have poor performance on such shuffled patches.
- Evaluating the networks on texture-shape cue conflict images as in Figure 1.1(c) which possess two different labels, the texture label and the shape label. A network with more shape bias will classify these images based on their shape labels while on the other hand, a texture biased CNN will categorize them according to their texture labels.

Upon evaluation, it is shown that the proposed approach of this thesis work encodes stronger shape bias. This shape biased network is further evaluated on ImageNet-C corruptions to verify if the improved shape bias helped in improving its corruption robustness. However, no performance improvement is observed on corruptions. On the contrary, the approach utilizing stylized images [1] for enhancing the shape bias of CNNs has shown improved robustness on corruptions as well. Moreover, the reason for this robustness improvement has

been attributed to its enhanced shape bias. Therefore, this thesis work further focuses on identifying the image factors that actually contributed to improved corruption accuracies on stylized images. Figure 1.2 gives a pictorial representation of the aforementioned steps. Together with these contributions, an additional insight, which suggests the possibility of improving the corruption robustness of a pretrained network by tuning only its normalization parameters without altering its feature extractors, is discussed.

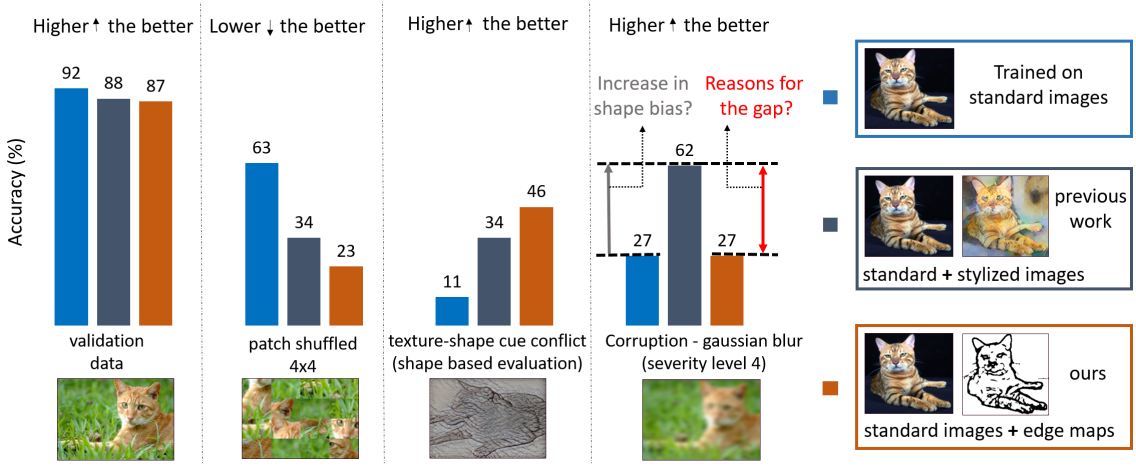

Figure 1.2: Pictorial demonstration of the activities involved in this thesis work.

1.4 Thesis Structure

The thesis report is organized into following chapters.

Chapter 1 This chapter provides a short introduction to the thesis work along with the motivation behind this thesis and a brief overview of the proposed solution.

Chapter 2 This chapter presents a short description of Convolutional Neural Networks (CNNs) with additional information on specific topics required to understand this thesis work.

Chapter 3 This chapter gives an overview of the contemporary research works related to this thesis work on topics like behavioral analysis of CNNs, improving robust representation of CNNs and enhancement of corruption robustness of CNNs. In the end, it describes an existing edge detection approach utilized in this thesis work for extracting the edge maps.

Chapter 4 This chapter describes in detail the different dataset variants considered and also explains a method called *style randomization* introduced in this work to get rid of the texture bias in CNNs.

Chapter 5 This chapter provides a deeper insight into the different evaluation strategies

used in this work to verify the shape bias of CNNs. Further, it also discusses the validation method utilized for evaluating corruption robustness.

Chapter 6 This chapter illustrates the different experimental settings followed in this work so that the results are reproducible. It then illustrates the results of various experiments conducted. Besides, this chapter also gives a detailed analysis of the results of each experiment.

Chapter 7 This chapter concludes the thesis work by summarizing it and discusses possible future research activities that could extend the scope of current work.

2 Background

The fundamental information required for understanding this thesis work are discussed in this chapter.

2.1 Convolutional Neural Networks

Convolutional Neural Network (CNN) is a specific type of deep neural networks used for processing image inputs. CNNs are feed-forward neural networks with a large set of learnable parameters. These parameters get updated in the training process with the help of the backpropagation algorithm. During the forward pass, the network generates outputs which are then compared against the ground truth data and the deviations are measured using a loss function. This loss is then backpropagated to all layers of the network while the learnable parameters get updated accordingly. The formulation of loss function depends on the nature of the task involved. For example, a cross-entropy loss is generally used for image classification tasks. CNNs consist of various components like convolutional layers, pooling layers, normalization layers, fully connected layers and non-linear activation functions. These components are briefly described below.

Convolutional layers: Convolutional layers form the basic building block of a CNN. They carry out convolution operations using learnable filters of specific kernel size. The filter size is usually much smaller than the size of input images. Such filters are convolved over the entire image region using the sliding window technique. Using several distinct filters, a convolutional layer extracts different features in an image while preserving the spatial locality of the content. These filters are hence termed as feature extractors. These convolutional layers which involve sparse connections with a reduced number of learnable parameters are often beneficial for image manipulation tasks.

Pooling layers: They are generally used to reduce the spatial dimensions of the activation maps resulting from convolutional layers without losing any essential information. Pooling helps to speed up the computation as only the important data among a local neighborhood will be considered. Different pooling methods like max-pooling, mean-pooling and sum-pooling exist among which max-pooling is commonly used. Max-pooling retains only the maximum value within a local neighborhood.

Normalization layers: Normalization layers are used to standardize the inputs of the following layers by rescaling the outputs of the current layer to zero mean and unit variance. Such layers are useful for accelerating the training process. However, in order to preserve the expressiveness of CNNs, they are typically used with learnable scale and shift parameters. More details on different normalization techniques are discussed in the following subsection.

Fully connected layers: For Multilayer Perceptron, typically all the layers are fully connected. In fully connected layers, all the neurons of the previous layer are connected to every single neuron in the current layer. Hence, they have more learnable parameters when compared to convolutional layers. Generally, in CNN architectures, the last few layers remain fully connected.

Non-linear activation functions: They introduce non-linearity in the network and are required in each layer of a CNN. Without any non-linear activation function in a network, no matter how deep the neural network is, its output will always be linear. Several such activation functions like sigmoid, Hyperbolic tangent (tanH), Rectified Linear Unit (ReLU), etc., are used among which ReLU is the most popular one. The output of ReLU activation is x if the input x is positive and otherwise 0. There exist additional variants of ReLU namely Leaky ReLU, Parametric ReLU, etc. Though they are similar to ReLU activation for the positive input range, their key difference to ReLU is that they also have a small slope for the negative values of input.

2.1.1 Normalization Techniques

In general, the normalization layers are placed before the non-linear activations for each hidden layer of a CNN. There exist different normalization techniques and Figure 2.1 depicts a pictorial representation of few such methods. Each subplot in Figure 2.1 shows a feature map tensor in which N is the batch axis, C is the channel axis, and (H, W) are the spatial axes. Highlighted pixels in blue illustrate that these pixels are normalized using the same mean and variance values that were computed by aggregating them.

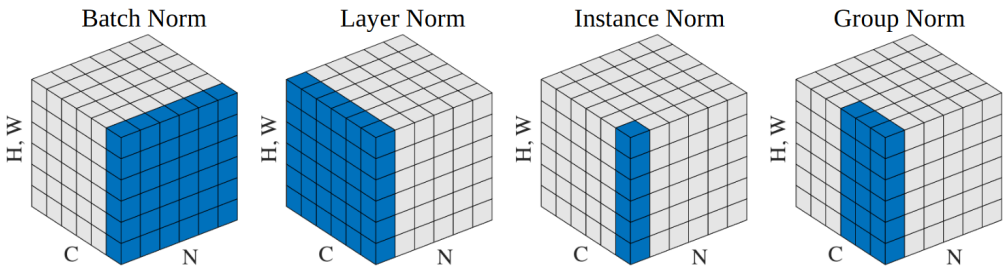

Figure 2.1: Different Normalization techniques [3].

General notation for normalization of an input x can be written as follows:

$$x_{normalized} = \gamma . \left(\frac{x - \mu_t}{\sigma_t} \right) + \beta \tag{2.1}$$

where μ_t and σ_t represent respectively the mean and the variance values, which are computed differently for different types of normalizations. Further, γ and β of Equation 2.1 are the learnable scale and shift affine parameters of the normalization function. These parameters are used in all kinds of normalization techniques.

Batch Normalization: In Batch Normalization (BN) [15], the network activations are normalized across a mini-batch of a specific size. This means that the channel-wise statistics computed across the mini-batch is used for normalizing the activations of each example in that batch. Thus, μ_t and σ_t of Equation 2.1 corresponds to $\mu(\mathcal{B})$ and $\sigma(\mathcal{B})$ in BN, where $\mathcal{B}$ represents the mini-batch. Batch Normalization aids in faster convergence of the training process. This was initially attributed to the fact that BN reduces the internal covariate shift that occurs in the input of each layer during the training process due to the weight updates. However, Santurkur et al. [16] has shown that this hypothesis doesn't stand true always. Further, it claimed that BN makes the loss landscape considerably smoother which ultimately aids in better performance. In addition, since BN introduces some noise in the training, it provides regularization effects to some extent. Though BN shows incredible results than other normalization techniques, its dependency on the mini-batch size is a potential issue and it exhibits poor performance for very small batch sizes. Another issue with BN is that it cannot be applied for Recurrent Neural Networks (RNNs) as the statistics of the activations vary during each time-step.

Layer Normalization: The key difference between Layer Normalization (LN) [17] and BN is that LN normalizes the activations of each example over the entire channel. Hence, LN is independent across each example and is unconstrained by the batch size used. Considering Equation 2.1, μ_t and σ_t in LN corresponds to $\mu(\mathcal{C})$ and $\sigma(\mathcal{C})$, where $\mathcal{C}$ represents the channel. LN is shown to perform well on RNN tasks.

Instance Normalization: Instance Normalization (IN) [18] is similar to BN except that it normalizes the channel-wise statistics across each example. Thus, in Equation 2.1, μ_t and σ_t are substituted by $\mu(x)$ and $\sigma(x)$ for IN, where x represents a single input image. IN could be considered as Contrast Normalization that makes the network agnostic to contrast variations of the input images. In feature space, an IN layer encodes the style information of images and hence could be effectively used for style transfer techniques. Style transfer, in general, refers to the stylization of an image content using the statistics of a style image.

Adaptive Instance Normalization (AdaIN) is an arbitrary style transfer technique that utilizes Instance Normalization for performing the style transfer [4]. IN results in style normalization as it normalizes the feature statistics that carry the style information of images. These style normalized feature statistics of the content image are then adjusted to the statistics of the style image which results in style transfer. To be precise, when x represents the content input and y is the style input, AdaIN adapts the channel-wise mean and variance of x to match with the statistics of y as follows:

$$AdaIN(x, y) = \sigma(y).\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y) \tag{2.2}$$

Here, $\mu(x)$ and $\sigma(x)$ represent the channel-wise mean and variance of the content input x, while $\mu(y)$ and $\sigma(y)$ are the channel-wise mean and variance of the style input y. For AdaIN, the styles could be arbitrary and need not be previously seen by the network. As in Figure 2.2, Huang et al. [4] utilized a VGG-19 network for AdaIN style transfer where the first few layers encode the content as well as the style features. They are followed by an AdaIN layer that performs style transfer in feature space as per Equation 2.2. Finally, a decoder produces the stylized output images.

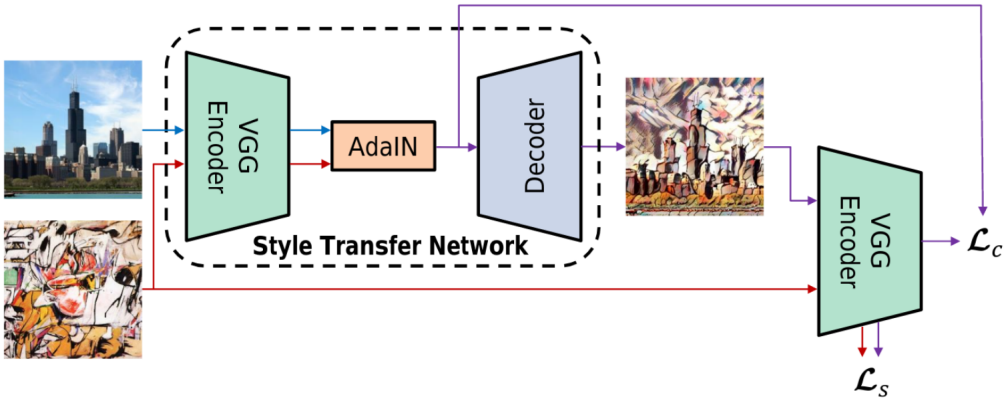

Figure 2.2: AdaIN style transfer network [4].

AdaIN also allows to control the degree of style transfer using a parameter α that ranges between 0 and 1. When α=0, it represents no stylization and the feature statistics of the content image are preserved as such. Increasing the value of α increases the amount of stylization. Maximum stylization can be achieved with $\alpha = 1$. Figure 2.3 shows an example of AdaIN style transfer where the leftmost image with α=0 represents the content image and the rightmost one is the style image. Other images represent the stylized outputs at different values of α. AdaIN is not only an efficient style transfer technique but also known to be faster than most of the existing methods.

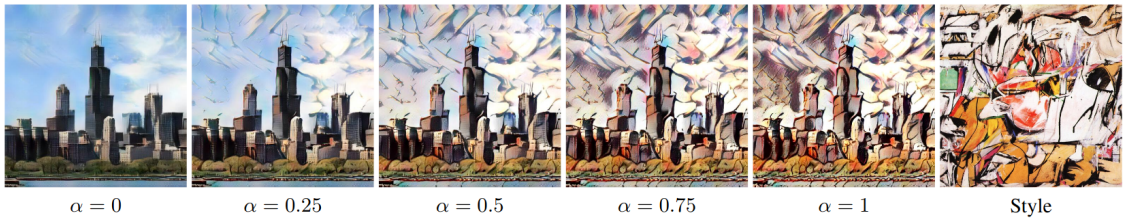

Figure 2.3: An example of AdaIN style transfer with varying degrees of stylization [4].

Group Normalization: Group Normalization (GN) [3] applies normalization for each example using the statistics of a group of channels. It can be considered as an interpolated technique between LN and IN. This is because LN normalizes each example over the entire set of channels while IN normalizes each example across individual channels. In GN, μ_t and σ_t of Equation 2.1 has to be replaced by $\mu(\mathcal{G})$ and $\sigma(\mathcal{G})$, where $\mathcal{G}$ represents a group of channels for an example x. Since GN is independent for every example, it works effectively than BN for smaller batch sizes.

Weight Standardization: Like other normalization techniques, Weight Standardization (WS) [5] is also introduced for accelerating the training process of CNNs. Among the previously discussed methods, BN outperforms in case of larger batch sizes but has poor performance for micro-batch training. Other normalization methods that have no dependency on batch size show performance benefits over BN only for smaller batch sizes. WS has been introduced to addresses this trade-off.

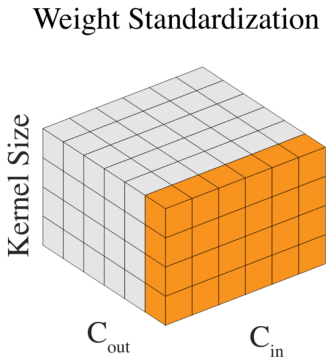

Figure 2.4: Weight Standardization [5].

As the success of BN is related to its smoothness effect on the loss landscape [16], WS also smoothens the loss landscape by standardizing the learnable weights of the convolutional layers itself. This is different from the other normalization techniques which focus on the network activations. Figure 2.4 represents a weight tensor, where C_{in} refers to the number of input channels and C_{out} refers to the number of output channels in a layer. Using the

statistics across each output channel, WS reparametrizes these weights. Qiao et al. [5] has shown that when WS is combined with GN, it outperforms BN irrespective of the batch size considered.

2.2 Robustness

While Convolutional Neural Networks (CNNs) show high-performance results on various tasks, their robustness is a topic of utmost importance. This is because it is crucial to ensure that the CNNs are robust enough to be deployed for practical applications. However, there exist different perspectives for the term robustness. Understanding and reducing the behavioral differences between human visual perception and CNNs is one such perspective. Although CNNs could match human-level performance in various image manipulation tasks, they are recently demonstrated to exhibit wide behavioral differences than that of humans. Those differences are described in the following subsections.

2.2.1 Understanding Shape Details

An object is composed of different features like shape, texture, color, contrast, brightness, etc. Among these, since shape is a robust feature, it acts as the most important cue for humans in recognizing objects [13]. When other local details like texture, color intensity, and contrast of the objects are modified, humans can still identify the objects correctly without any trouble. In a similar way, CNNs are also expected to focus on object shapes rather than other local statistics. But recent works [1, 2] have shown that CNNs are biased towards texture details for image classification tasks and therefore they fail when the texture information is modified. As stated already, addressing this gap between humans and CNNs is the main motive behind this thesis work.

2.2.2 Robustness to Common Corruptions

Corruptions represent stochastic image transformations motivated by real-world effects. The human visual system is robust and it cannot be fooled by corrupted images. Robustness against such transformations is essential for CNNs for their deployment in real-world applications. To verify this, Geirhos et al. [14] conducted experiments between humans and CNNs on different distortions such as noise, filters, contrast variations, and manipulations from the eidolon toolbox [19]. When the level of distortions is increased, a wide divergence in the error pattern between humans and CNNs are observed. Humans are shown to be more robust than CNNs in almost all the settings. Besides, Hendrycks et al. [12] introduced a benchmark dataset named ImageNet-C that consists of 15 commonly occurring corruptions at five different severity levels. These corruptions are grouped under the categories noise, blur, weather conditions, and digital transformations. When pretrained networks are evaluated on these corruptions, it has been observed that the CNNs are not robust against corruptions and

their accuracies drop rapidly with increase in severities. It has been further stated that the texture bias of CNNs is the reason behind their poor corruption robustness. Also, Geirhos et al. [1] has shown that increasing the shape bias of CNNs improves their corruption robustness. However, this hypothesis is clearly studied and analyzed in this thesis work.

2.2.3 Robustness against Adversarial Examples

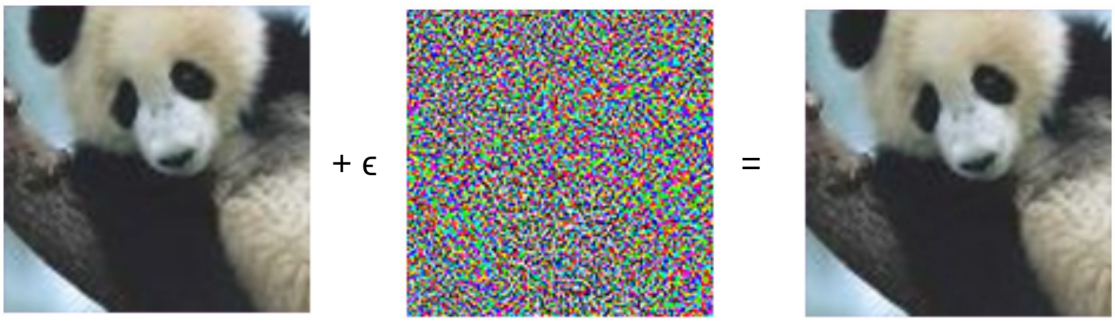

Figure 2.5: Adversarial example generation [6].

Szegedy et al. [20] has shown the existence of adversarial images that are generated by adding very small perturbations to the original images, which are imperceptible to human eyes. Such adversarial examples could mislead the CNNs to classify them as images from different categories rather than the actual ones with very high confidence. Several methods have been proposed over the years to generate such adversarial examples [6,21–23]. Figure 2.5 represents an adversarial image generated via Fast Gradient Sign Method (FGSM) [6]. It can be seen that when a small portion of the noise is added to a panda example, the resultant image has no visual difference. However, a pretrained CNN misclassified it as gibbon with more than 99% confidence. Robustness of CNNs against these adversarial examples is crucial as it pose direct questions on the ability of CNNs to perform safety-critical tasks. For example, devoid of this robustness, a CNN could misclassify a road sign when a small sticker is pasted over it. Though not under the scope of this thesis work, immense research works are carried out in this regard for improving the network’s robustness against adversarial attacks.

3 Related Work

This chapter provides the reader with an overview of already existing research works that are related to the thesis. It has the following sections:

1. Studying the behavior of CNNs - provides details regarding the works that analyze the behavior of CNNs and its comparison against the human vision.
2. Enhancing robust representations of CNNs - summarizes different existing approaches that enforce CNNs to learn more robust features like shape.
3. Robustness against common corruptions - As the thesis work presents a detailed study of the correlation between shape learning and corruption robustness, this section illustrates previous works that focus on improving the robustness of CNNs against corruptions.
4. Edge detection - Shape learning presented in this thesis work is achieved using edge information and hence this section discusses in detail, an existing edge detection approach used for the same.

3.1 Studying the Behavior of CNNs

Recent works revealed that the Convolutional Neural Networks (CNNs), which achieved human-level performance in image recognition tasks, show a wide behavioral difference than humans. For example, the global shape of objects acts as the most significant cue for humans in recognizing them [13]. Baker et al. [2] conducted various experiments between humans and CNNs to understand if CNNs also recognize objects based on global shape details. Five different experimental settings are considered. In the first setting, image silhouettes that are filled with texture details of different objects are accounted. In the second experiment, glassy images are included. The third experiment is conducted on object outlines. The fourth experiment included object silhouettes, where the foreground and background details are filled with different colors. Considering these four experiments, CNNs had poor performance when compared to humans. Moreover, humans did not have problems in recognizing these objects. This shows a strong divergence in behavioral patterns of CNNs from humans. Noticeably, in all these settings the images are devoid of the surface statistical cues like texture information, color gradients, etc. This further implies that such surface details play an important role in CNNs for recognizing objects. Among the considered settings, CNNs had better results for object silhouettes and it shows that CNNs are not completely devoid of the shape details.

Figure 3.1: Example images for each experimental setting considered in [2].

In order to further investigate whether the shape information included in CNNs are local or global to an image, a fifth experiment is conducted by considering the image silhouettes that are correctly classified by the CNNs. Two kinds of modifications are considered with those images. In the first part, the global object shape is disrupted by splitting the images into different patches and randomly shuffling them before joining. Surprisingly, the network still predicted those images with good accuracy while due to the lack of global shape, humans failed to recognize them. In the second part, only the local shape information of the image silhouettes is disrupted without disturbing their global shape. Such images are easily predicted by humans whereas CNNs have a hard time in recognizing them. These settings therefore demonstrate the contradicting behavior between humans and CNNs. Humans need global shape information for classifying objects while CNNs focus only on the local shape details. Figure 3.1 contains example images from each of the above discussed experimental settings. These results conclude that unlike humans, CNNs focus on surface details such as textures for classification. In addition, the shape cues, if any that are focused by CNNs are local shape details while humans require global shape information for successful classification.

Geirhos et al. [14] conducted several experiments to identify the differences in robustness between humans and CNNs. Around twelve different image distortions like color and contrast variations; additive noises at different severity levels; low-pass and high-pass filters with varying standard deviations; and Eidolon perturbations with different reach values [19] are considered for this study. Different network architectures like VGG19 [24], ResNet-152 [25] and GoogLeNet [26] are utilized. Several human participants are involved in this experiment and their responses are averaged. Human visual system is observed to withstand most of the distortions far better than any of the CNN architectures. Further, a wide divergence in the error pattern between humans and CNNs is observed. With stronger distortions, in most cases, all the CNNs are shown to be biased towards one or two categories of classification while human responses are still distributed among all the considered image classes. Further, when the CNNs are trained with augmented data from distortions, they outperformed humans on those particular distortions. However, training a CNN with data from one distortion did not show any accuracy improvement on the other considered distortions. Similarly, when trained with data from two different distortions, the network shows greater accuracy improvement only for those two distortions. In order to achieve good performance on all distortions, training a CNN with data from all such distortions may not be a good idea as there could be unlimited possible distortions in practice. Nevertheless, humans show better results for any

distortion, including those that may be rarely seen in their lifetime. This shows the high generalization capability of humans. Achieving such high generalization is crucial for CNNs as well.

Geirhos et al. [1] considered different experimental settings to verify the texture bias of CNNs. Different CNN architectures like AlexNet [27], GoogLeNet [26], VGG16 [24] and ResNet-50 [25] are included. In addition, a considerable number of human participants are also accounted for the experiments so that a direct comparison between humans and CNNs is possible. Experimental settings included the original images, greyscale images, black image silhouettes with white background, edge maps extracted from canny edge detector [28], and image textures. Upon interpreting the results, original and greyscale images are well classified by both humans and CNNs. It has to be noted that these settings retain both texture and shape details within the images. Further, the texture images that retain only the texture information from the objects are also well recognized by CNNs while a small performance drop is observed with humans on such images. However, the performance of CNNs dropped rapidly on image silhouettes and further significantly on edge images. These settings are devoid of texture details and they retain only the global object shapes. On the other hand, humans could manage higher performance on these images. These experiments support the claim of Baker et al. [2] that the texture details are more important cues for CNNs in classification. Figure 3.2 represents example images of the discussed experimental settings along with their predictions by the considered CNNs including the average predictions of human participants.

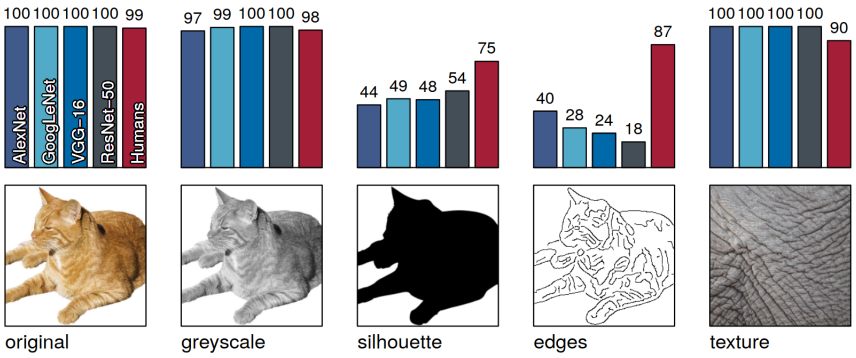

Figure 3.2: Different images accounted for the experiments of [1] along with their prediction accuracies by various CNN architectures as well as humans.

Apart from the discussed settings, Geirhos et al. [1] constructed a special kind of images called texture-shape cue conflict images. Such images are formulated through an iterative style transfer technique [29] which transfers the style of one image to a different content image. The style images considered for this technique are from the texture dataset while the content images are taken from the original images. Those images that are correctly classified by all the considered CNNs are chosen to be the style and content images for this style transfer. The resultant images from the style transfer include the shape of one object while their encoded

texture details depict a different object. Hence, these images have two labels namely, the shape label and the texture label. Such a setting is aimed to give more details about texture vs shape bias between CNNs and humans as their predictions on these cue conflict images would be either texture or shape biased. Supporting the previous claims, CNNs classified cue conflict images based on their texture while humans classified them according to their shape. This concludes that the CNNs perform successful image manipulations through their reliance on texture details and they have very little to do with the global object shapes which humans utilize for classification.

In addition to the above claims, there exist a reasonable number of works that demonstrate the texture bias of CNNs. Brendel et al. [30] introduced a variant of ResNet-50 architecture called BagNet which was inspired from the Bag-of-Feature (BoF) models that classify images based on the counts of a set of local image features. The receptive field size of such models is very limited and thus they can't consider the spatial relationship between features for classification. Surprisingly, BagNet architecture showed high performance similar to the state-of-the-art CNNs in image classification tasks. In addition, the authors had shown that there exists close similarity between the decision making capabilities of standard CNNs and BagNets [30]. These results suggest that the CNNs make their decisions based on local statistical regularities rather than the global image features.

Further, astonished by the contradicting facts that the CNNs could generalize well to unseen test data but their performance drops significantly with adversarial examples, Jo et al. [31] investigated the tendency of CNNs in learning surface statistical regularities that are present in both training and test datasets. To do so, new datasets are constructed which had exact similar representations as the original images except that they had different surface statistical cues. For creating such datasets, Fourier filtering with random and radial masks in the frequency space are applied and those changes in the resultant images are imperceptible to human eyes. Upon evaluation, the neural networks trained on such Fourier filtered images generalized well to the original unfiltered dataset. On the other hand, the networks that are trained on natural images could not generalize to the filtered datasets [31]. This implies that the standard networks develop a tendency to learn the weak surface statistical regularities within the images and hence could not generalize to those datasets that differ in such cues.

3.2 Enhancing Robust Representations of CNNs

In order to improve the shape bias of existing CNN architectures, Geirhos et al. [1] introduced a stylized dataset named Stylized ImageNet (SIN). This dataset is obtained through AdaIN style transfer [4], where the content images are taken from ImageNet1000 (IN) dataset and the styles are considered from kaggle's Painter by Numbers dataset. This style dataset has around 70,000 arbitrary collections of paintings. The resultant SIN dataset bears the content of IN data while their encoded styles are from paintings. Therefore, the texture details of SIN dataset are no more meaningful and cannot be utilized by the CNN for classification. Figure

3.3 represents a content image on the left while the stylized versions of it using different paintings are portrayed on the right.

Figure 3.3: Illustration of image stylization [1].

CNNs trained on stylized dataset forcefully rely on robust details like shape for successful classification. On evaluating such networks with texture-shape cue conflict images [1], the results seem to be more shape biased. On the other hand, a standard network trained on IN dataset has more texture biased results. Further, when a network is initially trained on SIN dataset and then finetuned using both IN and SIN images, it achieves similar validation accuracy as standard networks while its decisions are now biased towards shape details. Such shape biased networks are shown to be helpful with transfer learning tasks. In addition, these networks show improved robustness against various corruptions of ImageNet-C dataset [12]. The reason for this improved corruption robustness is attributed to the improved shape bias of the network. To be precise, since corruptions are observed to distort the local surface cues like texture of an image, a standard CNN with more texture bias cannot withstand them. On the other hand, since the network trained on SIN dataset has increased shape bias, it is more stable towards such corruptions and shows improved robustness. This thesis work includes a detailed study on this hypothesis where we show that there exists no clear correlation between the shape bias of a network and its corruption robustness. Further, with our experiments it can be understood that in case of SIN, the stylization of IN data acted as a data augmentation technique and it helped with improving the corruption robustness. However, additional works that employ different augmentation techniques on the dataset to improve corruption robustness of CNNs are discussed in detail in the next section.

Luo et al. [7] proposed a different approach to improve the robust representations in CNNs by introducing defective convolutional layers. This work is essentially aimed at improving the robustness of CNNs against adversarial examples. As discussed earlier, adversarial examples are created by slightly modifying the surface details of the original images. Recent findings [1, 2] have shown that CNNs rely on such local surface statistical regularities for classifying the images. Hence, Luo et al. [7] proposed a network architecture that has less reliance on texture information and has focus towards robust features like shape to gain adversarial robustness. For this purpose, defective neurons are introduced in the network. A neuron is termed defective if its output is always fixed to zero regardless of its input value. A

convolutional layer with such defective neurons is termed as defective convolutional layer. Luo et al. [7] claimed that such defective layers with enough defective neurons force the network to focus on robust features. This is attributed to the fact that the values of defective neurons vary a lot than their spatial neighbors and hence the local textural information cannot be easily extracted by the network. A CNN architecture which contains defective convolutional layers is termed as Defective CNN [7].

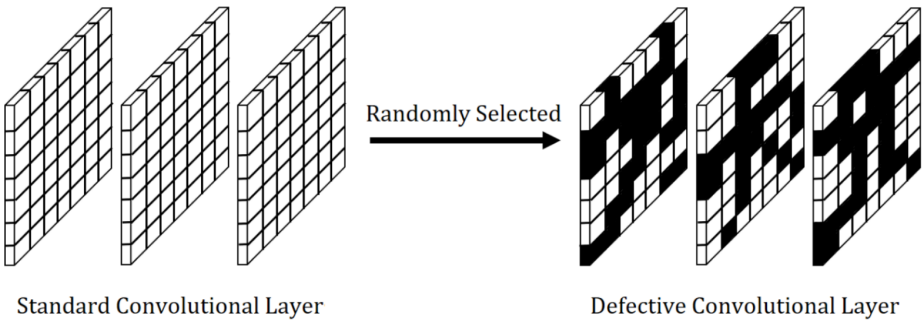

Figure 3.4: Defective convolutional layers [7].

In order to have defective convolutional layers, a fixed defective mask is randomly sampled using Bernoulli sampling at certain keep probability. This mask is then used for element-wise multiplication on the activation maps that finally results in a defective layer. Figure 3.4 illustrates the formulation of defective convolutional layers where the boxes filled in black represent those randomly selected defective neurons. Such defective neurons, when introduced in the initial layers of a network are shown to give better results. Further, the keep probability is held at a lower value which results in more defective neurons. Since the output of the defective neurons that are present in the initial layers of the network is zero, the convolutional kernels at the following layers cannot accurately extract the texture features. Though this method seems similar to the dropout strategy [32] which is mainly used as a regularization technique to prevent overfitting, there exist certain differences. In dropout, during the training process, some neurons are randomly dropped at each step. By doing so, the network cannot overfit to the training samples. However, all the neurons are accounted for validation. In contrast to this, the defective masks in Defective CNNs are used during both training and testing. Also, the values of these masks are fixed and they never change. The defective CNNs are evaluated on shuffled image patches and are shown to encode more shape features than a standard network. Shuffled image patches are created by splitting an image into different patches and then rejoining them after randomly shuffling those patches. Hence, these shuffled images lack the global shape while their texture details are still preserved. Defective CNNs have lower accuracies on such images than a standard network. This implies that defective CNNs rely less on local features like texture and more on the global shape details of the images. The defective network is shown to be more robust against different black box attacks. Further, the adversarial examples generated by defective CNNs are claimed to possess changes in the semantic details and so, they may even fool humans. Nevertheless, there are certain

drawbacks with this implementation like the presence of so many defective neurons in a network will directly impact its validation accuracy and also instead of random selection, following some systematic ways to choose defective neurons could be more effective.

Nam et al. [8] proposed a network for domain adaptation tasks and their methodology is related to learning robust features. It is well known that CNNs trained on images from certain domains do not possess the capability to adapt themselves to different domain data [33]. The reason attributed to this is again the network's reliance on local image styles for classification. Hence, Nam et al. [8] claimed that enforcing a network to focus on more reliable features like shape would make it style-agnostic and may ultimately help with domain adaptation. To achieve this, a Style-Agnostic Network (SagNet) is proposed that includes three main techniques namely style adversarial learning, style blending, and style consistency learning. Figure 3.5 shows the framework used for style adversarial learning.

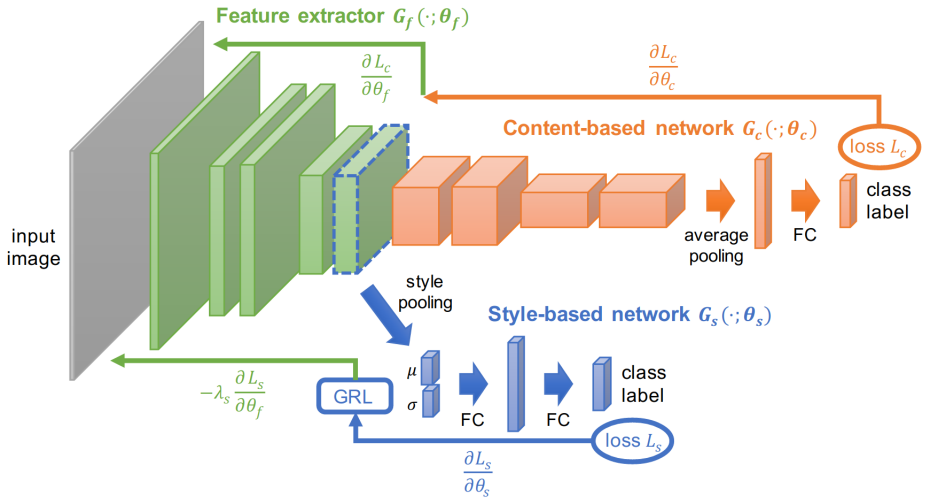

Figure 3.5: Style adversarial learning framework [8].

In style adversarial learning, the reliance of a network towards the local statistical cues is reduced by using an additional classification head which classifies the images based on channel-wise statistics (mean and variance) of an intermediate layer. To be precise, the original classification network, termed to be content-based network, makes the classification out of the complete network architecture. On the other hand, a second network named style-based network that shares the initial layers of the content-based network performs classification using the feature statistics of the final shared layer. These feature statistics encode the style information of the images. Since the aim is to force the network not to focus on such statistics for classification, a gradient reversal layer [34] is used to backpropagate the loss from the style-based network to the shared initial layers. Therefore these shared layers have to focus on other robust features for successful classification.

In the interest of further making the network agnostic to styles, a technique named style blending is introduced. With style blending, the style information encoded in the feature

maps are randomized during training. This is carried out by interpolating the feature statistics between the considered image and another randomly chosen image in the same batch. Since these feature statistics encode the style information, such an interpolation introduces randomness in the image style. The ratio between the preserved and randomized statistics is decided using a parameter α with its value ranging between (0,1) chosen from a uniform distribution. Equation 3.1 [8] shows how style blending is applied on a feature map X_i in feature space, by interpolating the statistics of example i with a randomly chosen example j. It has to be noticed that, a similar approach with slight modification than style blending is implemented in this thesis work to reduce the texture bias of our network.

$$
\begin{aligned}
X_i := \hat{\sigma}_i . \left(\frac{X_i - \mu_i}{\sigma_i} \right) + \hat{\mu}_i, \\
\textbf{s.t.} \quad \hat{\mu}_i = \alpha . \mu_i + (1 - \alpha) . \mu_j \\
\hat{\sigma}_i = \alpha . \sigma_i + (1 - \alpha) . \sigma_j
\end{aligned}
\tag{3.1}
$$

The third technique used in SagNet named style consistency learning introduces a consistency loss between two prediction vectors, one normalized using mini-batch statistics while the other one normalized using global moving-average statistics. Using different feature statistics for normalization allows style perturbation on latent spaces [8]. SagNet, implemented using the three techniques described above, is tested on a dataset named DomainNet [35], which contains images from different domains. The results imply that this network adapts well to data from unseen domains.

Wang et al. [36] also addressed the issue of CNNs learning surface statistical cues in an image, by introducing a method called Patch-wise Adversarial Regularization (PAR) that penalizes their local predictive power. In PAR, besides a standard classifier that performs original classification at the last layer of the network, it includes several side classifiers as well. These side classifiers jointly referred as patch-wise classifiers are applied at each spatial location of a lower layer in the network. The goal of PAR is to simultaneously perform two tasks. At first, it has to minimize the loss of the standard classifier. Alongside, it should restrict the side classifiers from making correct predictions. This is achieved using the reverse gradient technique [34] which performs gradient reversal while backpropagating the loss of patch-wise classifier to the shared initial layers. Hence, the predictive power of the local representations in those initial layers will be penalized while simultaneously, these layers are forced to focus on some robust features to ensure the correct classification of standard classifier. This implies that the classification network discards the local signals such as texture and color information in the images and focuses on the global structure of the image. The evaluation is carried out by using data from different domains since the cross-domain data possess similarities in the shape while varying a lot with the distributions. The authors used AlexNet as the baseline architecture and shown that PAR improves cross-domain accuracy. In addition to the existing datasets for domain adaptation tasks, this work introduced ImageNet-Sketch data that can be used for validating the cross-domain performance of ImageNet trained models.

A recent work by Hermann et al. [37] explored the reasons behind the texture bias of CNNs. They verified if it could be related to the network architecture or the type of dataset or because of the training procedure. Geirhos et al. [1] had earlier shown that the networks learn shape information when trained using certain augmented data. This implies that CNNs also possess the capability to learn shape details. However, it has to be examined whether CNNs can readily learn texture than shape. To analyze this, specific datasets that possess both texture and shape labels are utilized. They include texture-shape cue conflict images [1] where the shape and texture details are blended from two different images; Navon dataset [38] which contains larger alphabets (shape label) that are formulated using multiple copies of a different smaller sized alphabet (texture label); ImageNet-C dataset where the original object class is considered to be the shape label and the corruption type is regarded as the texture label. CNNs are trained on the above datasets for 90 epochs where the instances of both texture and shape labels are fed. Different training settings are considered based on the percentage of training data used, which ranges from 5% to 100%. Hermann et al. [37] observed that CNNs required lesser training data to achieve higher shape accuracies when compared to texture accuracies. In addition, across all the datasets, CNNs learned shape details faster than texture details.

With the above experiments, it is evident that shape features are as easily learnable as texture features in CNNs. However, CNNs are shown to be texture biased. Hermann et al. [37] addressed this difference with two possible explanations. One reason is attributed to the most common way of augmenting the training dataset before being fed to the network. It is the use of random-crop data augmentation which samples a random portion of an image between the default range scale of [0.08, 1.0] with a random aspect ratio between [0.75, 1.33] which is then resized to 224x224 pixels. Such augmentation may result in a small portion of the image and hence will be devoid of the global shape. It is further claimed that, instead of random-crop, using center-crop augmentation could preserve global shape information of the image as it just performs central cropping of 224x224 pixels on the whole image. Different experimental settings are considered with these two augmentations and the results show that the model with center-crop augmentation possesses more shape bias than the one with random-crop. However, it has to be noted that center-crop doesn't act as a proper data augmentation technique as it produces the same result for a given image. The second possible explanation for the texture bias of an ImageNet trained CNNs is the selection of its hyperparameters. Hermann et al. [37] argued that the set of hyperparameters that maximize the validation accuracies need not necessarily help with shape or texture bias of the network. Hence, a series of settings are considered with a different choice of values for the learning rate as well as weight decay. The results indicate that higher values of learning rate and weight decay implied more shape bias on the networks. Also, highest shape accuracy is observed for the network with the highest learning rate whereas highest texture accuracy is observed for the one with the lowest learning rate. Though the explanations for texture bias of CNNs are not very well backed in this work [37], it gives a different intuition regarding the learning nature of CNNs.

Zhang et al. [9] investigated adversarially trained CNNs to verify if they learn features that

are different from standard networks so that they become robust against adversarial examples. As discussed earlier, adversarial examples possess visually imperceptible perturbations to the clean images and these perturbations impact the classification accuracies of CNNs. These adversarial examples are structurally no different from original images while only their local surface statistical features vary. There exist several approaches that focus on improving the adversarial robustness of CNNs [23, 39–41] and one such common method is called adversarial training [6, 22]. The basic idea of adversarial training is to generate adversarial examples for the network and then incorporate these adversarial images along with the standard images for training the network. Such CNNs are termed as Adversarially Trained CNNs (AT-CNNs) [9]. Inspired by the recent findings [1, 2] which show that the ImageNet trained CNNs rely on texture details for classification, Zhang et al. [9] interpreted the AT-CNNs to check if they focus on different features than texture. Both visual and quantitative evaluations are carried out and the results indicate that AT-CNNs focus on more robust features than texture for classification.

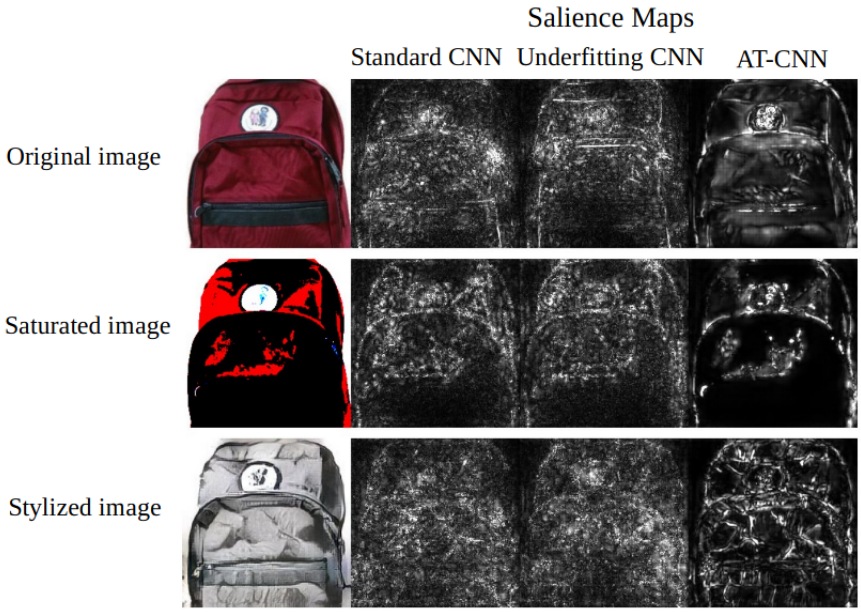

Figure 3.6: Salience maps from standard, underfitting and AT-CNNs on original, saturated and stylized images [9].

To conduct the experiments, Zhang et al. [9] considered TinyImageNet [42], Caltech-256 [43], and CIFAR-10 [44] datasets. For quantitative evaluation, two different image settings are accounted. The first setting included stylized [1] and saturated images where the global object structure is preserved but their texture details are modified. For the second setting, shuffled image patches that are devoid of the global shape but have unaltered texture information are considered. Further, for visual evaluation, salience maps of various images from standard, underfitting, and AT-CNNs are considered. The underfitting network is nothing but the standard CNN model whose validation accuracy matches the accuracy of AT-CNNs. These

salience maps could provide visual evidence whether these networks focus on different image features to perform classification. Figure 3.6 represent the salience maps from standard, underfitting, and AT-CNNs for an example image from the Caltech-256 dataset. The second and third-row correspond to the saturated and stylized version of that image along with their salience maps from different networks. It can be inferred from Figure 3.6 that AT-CNNs focus on more global features when compared to the two versions of standard CNN. Also, for the quantitative analysis, the standard network along with different adversarially trained networks [6, 23] is validated on stylized images, saturated images, and shuffled image patches. The results show that AT-CNNs have better accuracies than standard CNNs for stylized and saturated images. However, in case of shuffled patches, standard CNNs show better results than AT-CNNs. These results together indicate that the AT-CNNs learn more shape information and are less biased towards texture when compared to standard CNNs.

3.3 Robustness against Common Corruptions

As discussed earlier, common corruptions are stochastic image transformations motivated by real-world effects that influence the model robustness. Hendrycks et al. [12] introduced ImageNet-C dataset as benchmark data for corruption evaluations. Various approaches exist that are aimed at improving corruption robustness. Those include data augmentation [10, 45–47], self-training with more training data [48], novel architectures and building blocks [49, 50], and changes in the training procedure [51, 52]. Few existing techniques that involve different data augmentations to improve corruption robustness are explained in this section.

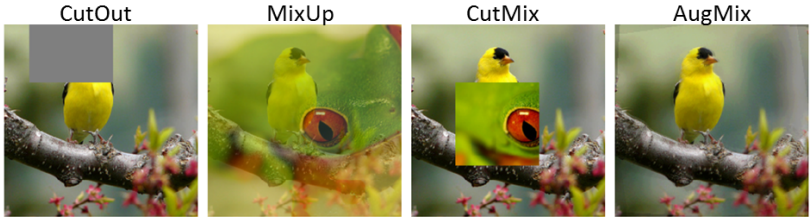

Figure 3.7: Visual comparison of different data augmentation techniques [10].

Lopes et al. [45] introduced a data augmentation technique called Patch Gaussian in order to improve the corruption robustness without affecting the clean data accuracy. In general, a network with high clean data accuracy has low corruption robustness. On the other hand, a technique that focuses on improving the corruption robustness could not achieve high validation accuracies on clean images. For example, CutOut [53] is an augmentation technique that takes a random patch of the input image and fixes it to a constant value. By doing so, the validation accuracy of CNN is improved. However, this doesn't help with corruptions. In contrast, a method called Gaussian augmentation [54] that adds some noise value, sampled from normal distribution, to each pixel in the image is shown to have improved corruption robustness but with reduced clean data accuracy. Patch Gaussian [45] is a technique that is

derived from these two methods. It selects a random patch in the image similar to CutOut but introduces Gaussian noise to that patch as in Gaussian augmentation. This augmentation technique is shown to have improved corruption robustness while at the same time, maintains high validation accuracy.

Similar to CutOut [53], Yun et al. [46] proposed a method called CutMix with slight improvisation. CutOut, which selects a random patch in the image and makes it to a constant value, has a drawback that it doesn't encode any useful information in the selected patch area. To overcome this, CutMix, which also selects a similar random patch in the image, fills this patch with data from another image. Proportional to the area of the patch size, ground truth labels are also mixed. Such an augmentation helps in improved validation accuracy and at the same time, has performance benefits with image localization and object detection tasks. In addition, it also improved the robustness of CNNs. Another technique called Mixup [55] performs data augmentation through convex combination of two images. It also shows improved corruption robustness without any drop in validation accuracy.

Hendrycks et al. [10] recently proposed a new data augmentation technique called AugMix with the aim of improving model robustness. AugMix addresses the issue concerning the difficulty of CNNs in generalizing to unknown distributions and its results imply improved accuracies on ImageNet-C distortions. It attained state-of-the-art results in both robustness and uncertainty estimations without any drop in clean data accuracy. AugMix uses different operations like translate, rotate, posterize, etc., from AutoAugment [56] and constructs augmentation chains that consists of one to three randomly chosen augmentation operations. Among the various operations in AutoAugment, those that are related to the ImageNet-C distortions are intentionally avoided in AugMix so that it remains disjoint from those corruptions. Each augmentation chain produces one resultant image and these images are then combined through a mixing operation that uses elementwise convex combinations. The weights used for mixing are randomly sampled. The resultant image is finally combined with the original image through another mix operation with random weights. Figure 3.8 depicts an example of the AugMix operation. Such augmentations result in an image that is not much different from the original one while at the same time incorporates diverse augmentations.

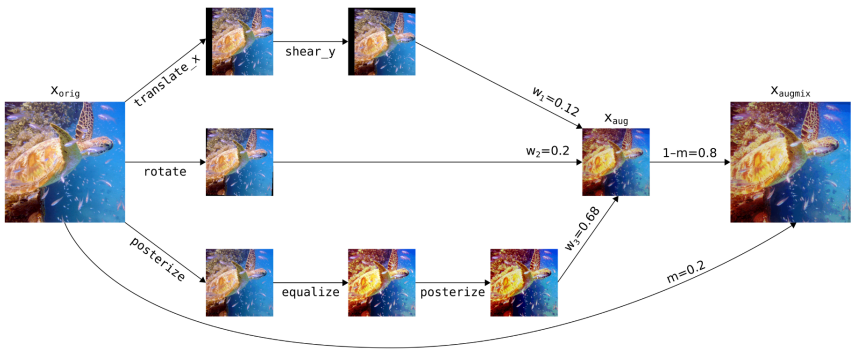

Figure 3.8: An example of AugMix operation [10].

In addition to AugMix, Hendrycks et al. [10] introduced a new loss term to smoothen the CNN response. Since the semantic meaning of the image is almost unchanged with AugMix, a loss that minimizes the Jensen-Shannon divergence between the original image and its augmented versions is introduced. AugMix together with this additional loss function is shown to outperform existing models on ImageNet-C distortions.

Stylized ImageNet (SIN) [1], which showed improved robustness against ImageNet-C corruptions, is also notably a data augmentation technique. Geirhos et al. [1] attributed this robustness improvement towards shape learning. However, in this thesis work, a detailed study is conducted by breaking down SIN into different factors and the results imply that style variations among the SIN dataset together with some preserved characteristics of natural images are responsible for the improved robustness against corruptions and not necessarily the shape. Moreover, using Stylized ImageNet dataset, Geirhos et al. [1] tried reducing the texture bias of CNNs which in turn improves the networks' shape bias. In this thesis work, the shape representation of CNNs is explicitly improved using edge information. In the next section, the method utilized for extracting such edge details is explained.

3.4 Edge Detection

Edges are fast gradient changes in the images and they represent the structure details of an object. There exist several algorithms for edge detection. Among conventional methods, canny edge detection [28] is a promising approach that employs a series of steps to extract the edge details in an image. However such conventional methods are less accurate and sometimes produce noisy results. There also exist feature learning based methods [57–59] which initially extract low-level features and then use sophisticated learning methodologies to predict edge pixels. However, they lack high-level representations. Recently, with the advancements in deep learning, many deep learning based approaches that use CNNs for edge detection are proposed [11, 60–62]. These methods are shown to extract edges with better quality when compared to the previous approaches.

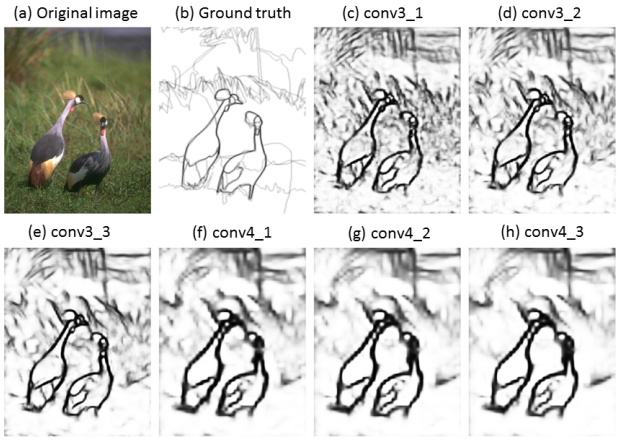

Figure 3.9: Edge outputs of varying details with RCF edge detection [11].

Among the deep learning methods, a recent state-of-the-art approach named Rich Convolutional Features (RCF) [11] is used in this thesis work for edge extraction. For edge detection in RCF, Liu et al. [11] utilized a fully convolutional network that is modified from VGG-16 [24] architecture. The convolutional layers of the network are grouped into five stages and RCF utilizes features from all these layers to produce the final edge map. However, in addition to the final output, it produces five side outputs from each stage of the network. The level of detail in these side outputs varies from fine to coarse between the initial and final stages of the network. Figure 3.9 visually depicts the edge outputs from different stages of the RCF network. Having such outputs with varying details is another prominent reason for choosing RCF among other deep learning approaches. For our shape based learning approach, a ResNet101 [25] based RCF edge detector with four side outputs that is pretrained on BSDS500 [59] dataset is utilized. Among the four outputs, the stage2 results contained essential details and hence are used for our purpose. Figure 3.10 gives a visual difference between the edges extracted using a canny edge detector and the edge maps from RCF network that is used in this thesis work.

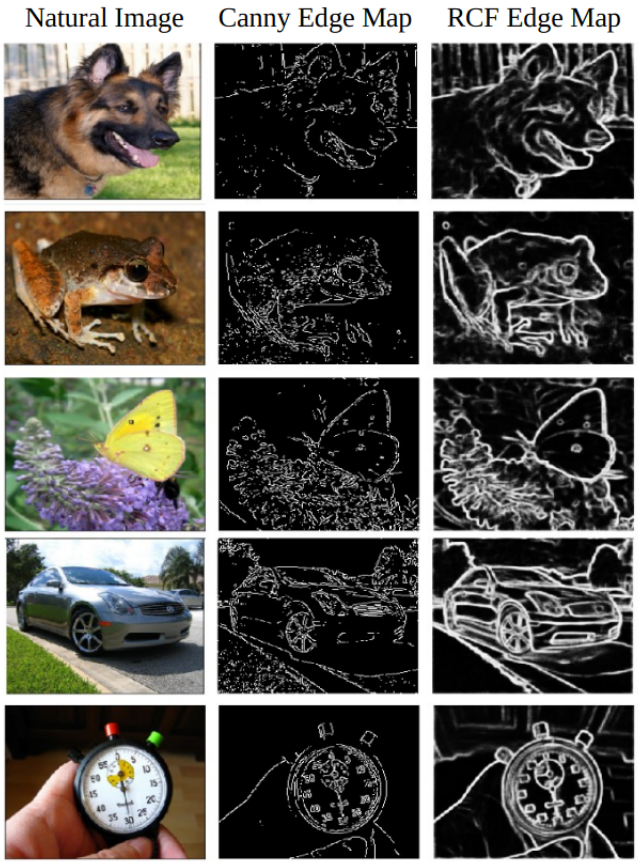

Figure 3.10: Canny vs. RCF edges.

4 Learning Shape Representations

This chapter gives detailed insight into the dataset and the techniques utilized in this thesis work for enhancing the shape representations of CNNs.

1. Dataset - discusses in detail the dataset utilized for various experiments included in the thesis work.
2. Learning Shape Representations using Edge Maps - provides details on the different edge map variants that are constructed for learning shape representations in CNNs.
3. Style Randomization - description of a technique which reduces the texture bias of CNNs in feature space.
4. Style-based Data Augmentations - discusses the different image variants that are constructed from style-based augmentations to understand the factors that influence corruption robustness.

4.1 Dataset - ImageNet20

Shape based learning has been explored in this thesis work using image classification tasks. Many popular datasets like CIFAR10, CIFAR100, TinyImageNet, ImageNet1000, etc., are available for the same. However, since CIFAR and TinyImageNet datasets include low resolution images, they cannot be properly visualized. Hence, the ImageNet1000 dataset which contains high resolution images is chosen for our experiments. Nevertheless, since it's a large dataset with 1000 classes, training a network on such a dataset may be costly in terms of time as well as resources. Thus, a subset of 20 classes is handpicked from the 1000 class ImageNet data and is used for conducting our experiments. It is termed as ImageNet20 and the acronym IN henceforth represents this dataset. It includes images from different categories namely natural images, man-made objects, automobiles, and edible items. Natural images included are animals, birds, and insects. Animal classes comprise `African Elephant, German Shepherd, Tabby Cat, Arabian Camel, Tailed Frog,` and `Scorpion`. Under birds, `King Penguin` and `Albatross` are included. `Fly` and `Sulphur Butterfly` are chosen from the insect category. Edible items include `Mushroom, Bell pepper,` and `Pretzel`. `Sports Car, Trolley Bus,` and `Life Boat` represent the automobile classes of the ImageNet20 dataset. Man-made objects considered are `Tea Pot, Stop Watch, Teddy Bear,` and `Fur Coat`. Figure 4.1 shows example images from each of the classes included in the ImageNet20 dataset.

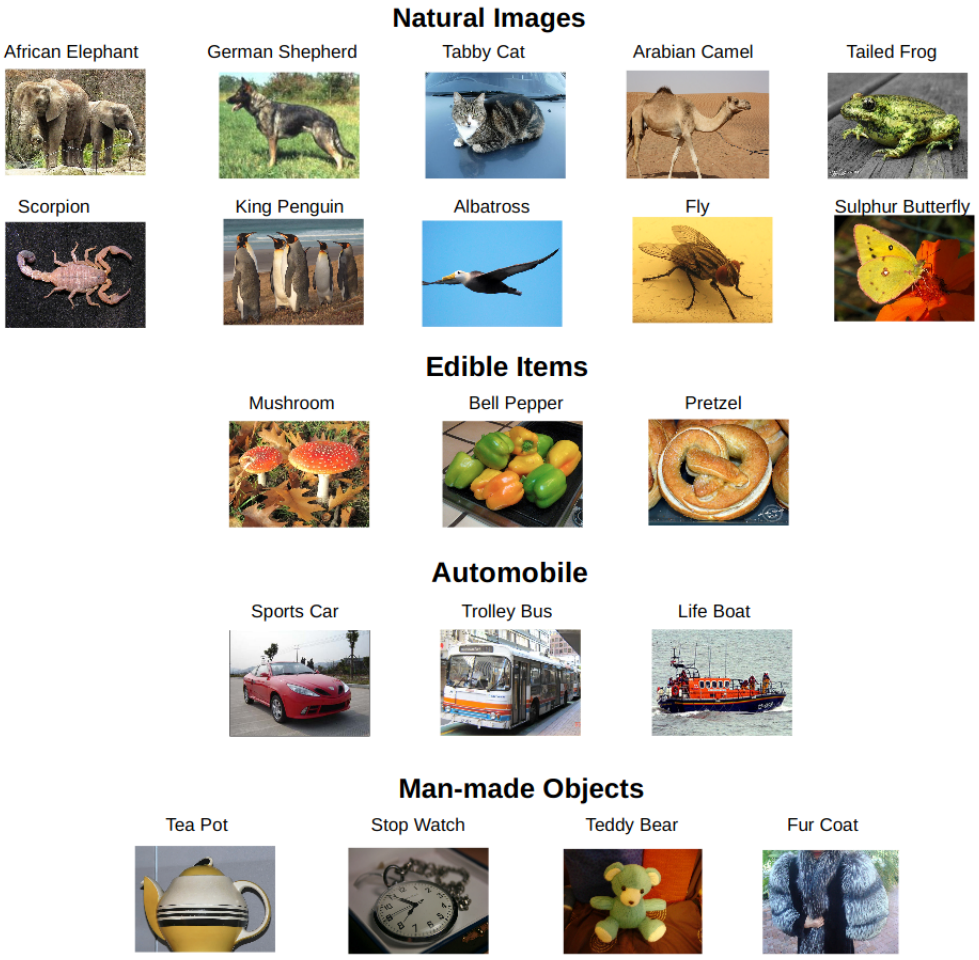

Figure 4.1: ImageNet20 dataset classes.

4.2 Learning Shape Representations using Edge Maps

This thesis work is aimed at improving the shape bias of CNNs thereby alleviating their high reliance on texture cues. Geirhos et al. [1] enhanced the shape bias of CNNs by training them on stylized images that don't retain original texture information. As the texture cues are no more relevant, such networks have to rely on more robust features like shape for correct classification. However, in this thesis work, the shape bias of a network is enhanced explicitly using the edge information extracted from the images. While stylization makes the texture cues less predictive, edge maps that are devoid of any other details enhance the shape representation of CNNs. It can be later inferred from the evaluation results presented in Chapter 6 that edge-based shape learning is efficient than Stylized ImageNet (SIN).

For a dataset of standard images, its corresponding edge dataset is constructed using the RCF edge detector as described in Section 3.4. As mentioned earlier, a ResNet101 based

RCF edge detector is used for our work and the side output of the second layer is considered for the generation of our edge dataset. Outputs of the RCF edge detector are single-channel edge maps which are then transformed into three-channel maps by duplicating the pixel values across the three channels. By doing so, these edge maps can be treated in a similar way as standard RGB images for training the CNNs. Edge maps from RCF network possess values between [0, 255] where the bright pixel values represent edges and the dark pixels are background details. Different variants of these edge maps are considered for our experiments. These variants are shown in Figure 4.2 and are explained in detail below.

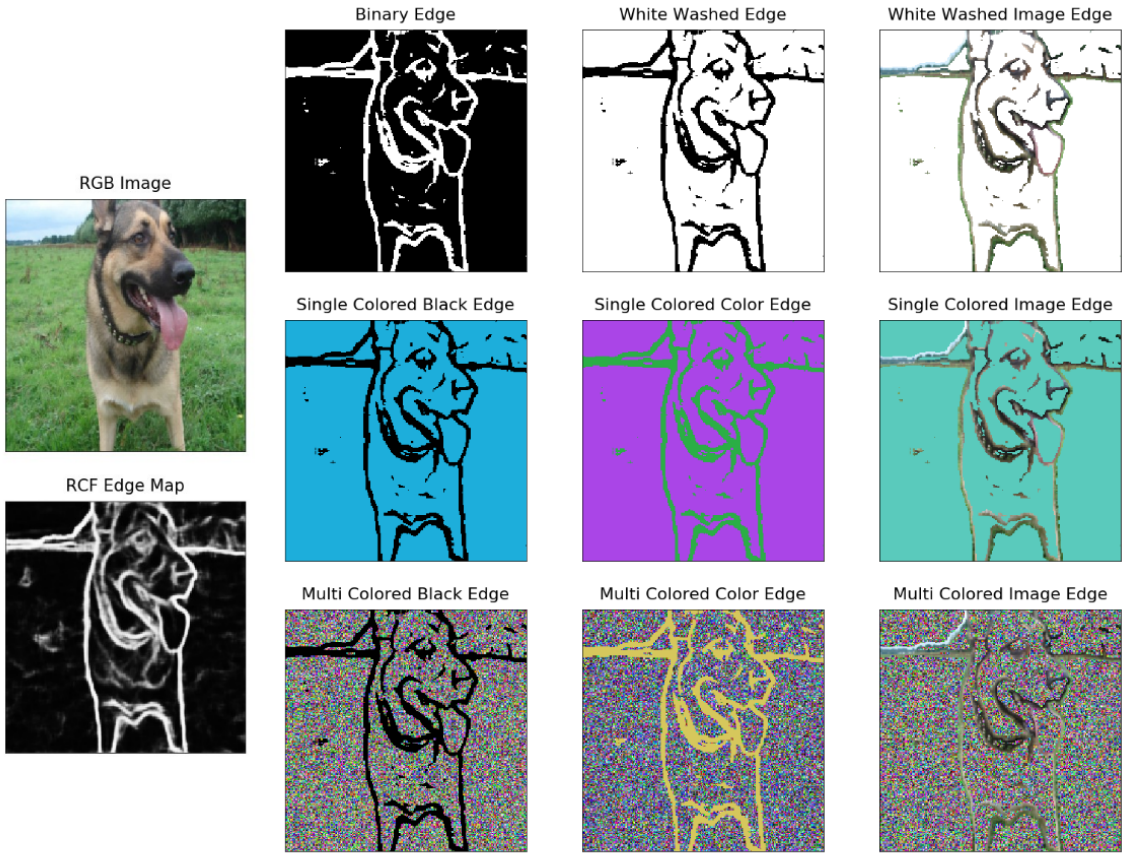

Figure 4.2: Different variants of edge maps considered in the thesis work.

Binary Edges: The first variant is Binary Edges, where the generated RCF edges are binarized to either 0 or 255. The threshold value for binarization is set at 128. The resulting binary maps possess white pixels for edges while the rest of the image is black.

White Washed Edges: Binary edge maps are reversed to form White Washed Edges. Here, the black pixels represent edges while the background information is white.

White Washed Image Edges: White Washed Image Edges are similar to White Washed Edges except that the edge pixels are directly considered from the images and are not black. The background information is retained to be white.

Single Colored Black Edges: As the name indicates, Single Colored Black Edges contain black edges while the background details are filled using a single randomly chosen color. In the interest of differentiating the background color from the black edges, the random color value is chosen within the range [10, 255].

Single Colored Color Edges: Single Colored Color Edges possess two different randomly chosen colors, one to represent the edges and the other to represent the background. For proper differentiation between edges and other pixels, a difference value of minimum 20 is guaranteed between the two chosen colors.

Single Colored Image Edges: In Single Colored Image Edges, the edge pixels are taken from the image while the background is chosen from a single random color ranging between [10, 255]. Since the edges are mostly observed to be dark pixels, a minimum pixel difference of 10 is considered for the background color.

Multi Colored Black Edges: Multi Colored Black Edges contain black pixels that represent the edges while every other pixel from the background is randomly chosen from the set of possible colors excluding black.

Multi Colored Color Edges: The difference for Multi Colored Color Edges from Multi Colored Black Edges is that the edges are represented using a single random color rather than black. The rest of the pixels carry random colors along with the constraint that they have a minimum difference of 20 from the edge color.

Multi Colored Image Edges: Multi Colored Image Edges have the edge information extracted from the images while each of the other pixels in the image is chosen randomly from [10,255].

Performance evaluation of these different edge variants is addressed in Chapter 6. Based on the results of these edge variants, White Washed Edges are chosen for shape based learning.

4.3 Style Randomization

The shape representations of CNNs can be improved in two possible ways. The first way is to explicitly enhance the shape bias while the other way is to reduce its texture bias. For example, when a dataset with explicit shape details is used for training a network, this can directly improve its shape representations. On the other hand, while training on natural RGB images, if one can prevent the network from learning local texture information, this will automatically enforce the CNN to focus on more robust cues like shape. While the edge maps are utilized for the first approach, a new technique is proposed in this work to reduce the texture dependency of CNNs. It can be noted that the Stylized ImageNet [1] also enhances shape representations of CNNs by reducing their texture dependency.

Inspired from the style transfer literatures [4, 63] which show that the style information of a dataset is encoded in the statistics of feature maps, namely the mean and the variance, a simple technique called *Style Randomization* is proposed. Style randomization is aimed at randomizing the feature statistics of a network so that it becomes style-invariant. This approach is similar to *Style Blending* [8] which randomizes the style information by interpolating the feature statistics between two different image samples within a mini-batch. But in style randomization, a slightly different procedure is used to modify the feature statistics. Instead of interpolating the feature statistics between the images, the statistics are completely randomized by randomly sampling the mean and the variance values from a uniform distribution. For example, considering X_i as the i^{th} feature map of an intermediate layer in the CNN, and μ_i and σ_i as the feature statistics of X_i, the style randomized feature map $\hat{X}_i$ is defined as follows:

$$\begin{aligned} \hat{X}_i := \hat{\sigma}_i \cdot \left(\frac{X_i - \mu_i}{\sigma_i}\right) + \hat{\mu}_i, \\ \textbf{s.t.} \quad \hat{\mu}_i \sim Uniform(-1, 1) \\ \hat{\sigma}_i \sim Uniform(0.1, 1) \end{aligned} \tag{4.1}$$

AdaIN [4] style transfer, which has been used for creating Stylized ImageNet [1], achieves style transfer by replacing the feature statistics of a content image with those of a style image. Similarly, here the texture details are randomized by modifying the feature statistics in feature space. It is shown in Section 6.2 that style randomization reduces the texture bias of a network and outperforms style blending, as the feature statistics are considered from a different distribution than the input images.

4.4 Style-based Data Augmentations

Besides improving the shape bias of CNNs, the resultant shape based network is evaluated on the ImageNet-C dataset [12]. The results of this evaluation motivated us to conduct

further experiments. This is because, Stylized ImageNet (SIN) [1] with higher shape results exhibits improved performance on ImageNet-C corruptions as well. Enhanced Shape bias is hypothesized to be the reason behind their robustness improvement [1]. However, even though the shape based network proposed in this thesis work demonstrates higher shape bias than SIN, it doesn't show any improvement with ImageNet-C distortions. This raises the question of whether the improved shape bias is the reason behind the corruption robustness of SIN. A detailed study is conducted in this regard and the results are discussed in Chapter 6. For this purpose, different image variants are constructed using style-based data augmentation techniques by splitting the factors that together constitute the stylized images of SIN dataset.

The key properties of SIN dataset include, the shape bias due to its style invariance, the possession of various styles from paintings, and certain preserved characteristics from natural images. It is probable that any of these factors separately or together would have resulted in improved corruption robustness. Though it is not feasible to completely disentangle these factors, they are segregated in possible ways and are used for our experiments. Figure 4.3 illustrates the different image variants considered. Different datasets are constructed based on these image variants for the analysis of corruption robustness. The newly introduced style variants are discussed below in detail.

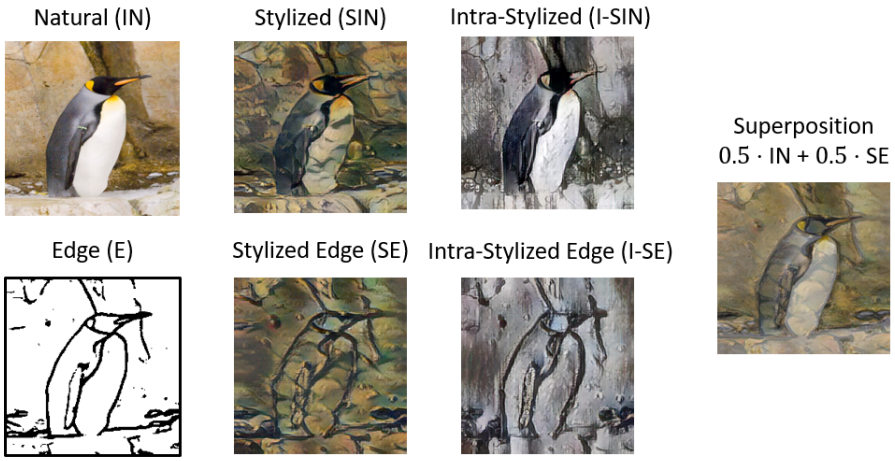

Figure 4.3: Different image variants considered for evaluating corruption robustness.

Role of shape bias: The role of shape bias in improving corruption robustness can be studied with the help of the Edge dataset as they carry only the shape details and are devoid of texture or other statistics of natural images. The *Edge dataset* is termed as **E** for further references.

Role of stylization: To understand the influence of stylization, a dataset that contains the style information as similar to the *Stylized ImageNet* (**SIN**) dataset has to be considered. However, it is not meaningful to consider only the paintings as they deviate from the network's training data. For this reason, the Edge dataset created from ImageNet20 is stylized using the same set of paintings that are considered for the stylization of SIN dataset. This stylized variant of Edge dataset is referred to as *Stylized Edges* (**SE**). Since the influence of shape bias can be already understood from the Edge dataset, Stylized Edges would help in analyzing the influence of stylization on corruption robustness.

Role of style distribution: As discussed, the styles considered for the creation of the stylized ImageNet dataset(SIN) are taken from a set of paintings. These paintings belong to a different data distribution than the original images and hence can be termed as out-of-distribution styles. To study the influence of such distribution variance in styles, additional datasets are considered where the styles are chosen to be random images from the ImageNet20 dataset itself. These are referred as in-distribution styles. Using such styles, two dataset variants are constructed. One variant, termed as *Intra-Stylized ImageNet* (**I-SIN**), is created by the in-distribution stylization of the ImageNet20 dataset. The second variant called *Intra-Stylized Edges* (**I-SE**) is created by stylizing the Edge dataset (E) using the in-distribution styles. Evaluation of these datasets would give insights on the importance of the distribution of style images.

Role of preserved natural image statistics: Visualization of the stylized image (SIN) from Figure 4.3 shows that the stylization however preserved some natural image statistics as well. To be precise, the color intensity at different segments of the natural penguin image from IN could still be observed in its stylized versions (SIN and I-SIN). However, Edge dataset (E) or its corresponding style variants (SE and I-SE) are devoid of such image statistics. In order to study the influence of such preserved image statistics on corruption robustness, a new dataset called *Superposition* (**SE+IN**) is constructed. As the name suggests, Superposition datasets are created by the interpolation between two different datasets. These include the natural images from ImageNet20 (IN) and the Stylized Edges (SE). During interpolation, the weightage among these two datasets is determined using a parameter α. So, α decides the amount of natural image statistics that will be preserved in the resulting Superposition image. For a given natural image (I_{IN}), and its corresponding Stylized Edge (I_{SE}), the Superposition image ($I_{\text{SE+IN}}$) can be defined as,

$$I_{\text{SE+IN}} := (1-\alpha) \cdot I_{\text{SE}} \; + \; \alpha \cdot I_{\text{IN}}, \quad \textit{where} \;\; \alpha \sim (0,1) \tag{4.2}$$

It is evident from Equation 4.2 that $\alpha = 0$ represents the Stylized Edge (I_{SE}) and $\alpha = 1$ represents the natural image (I_{IN}). At $\alpha = 0.5$, both I_{SE} and I_{IN} share equal contribution

towards the generation of $I_{\text{SE+IN}}$. For $\alpha < 0.5$, the resultant Superposition image will have less image statistics and more stylization effect. On the other hand, for $\alpha > 0.5$, the generated image will retain more natural image statistics than the amount of stylization. Figure 4.4 represents Superposition images constructed using different α values. Here, the left-most image represents the Stylized Edge and the right-most one is the natural image.

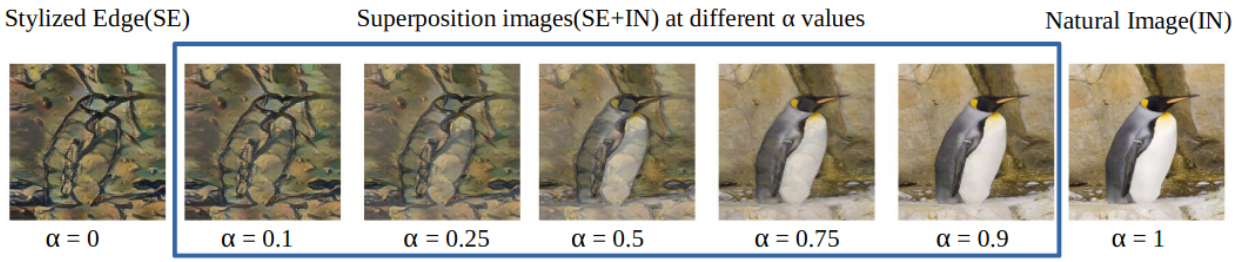

Figure 4.4: Superposition images at different values of α.

To clearly understand the role of shape bias in corruption robustness, different CNNs are trained on the above-discussed dataset variants and are evaluated on ImageNet-C distortions. The results along with their inferences are discussed in Chapter 6.

5 Evaluation Strategies

This chapter provides details regarding the different evaluation strategies employed in this thesis work to determine the shape bias and the corruption robustness of CNNs. The following are those strategies.

1. Shuffled Image Patches - to evaluate the shape bias of a CNN.
2. Texture-Shape Cue Conflict Images - for quantitative comparison between the texture and the shape bias of a network.
3. ImageNet-C Corruptions - to evaluate the network's robustness against common corruptions.

5.1 Shuffled Image Patches

The shape bias of a network is evaluated using Shuffled Image Patches in which the global shape details of the images are destroyed and only their local texture details are unaltered. A shape biased CNN is expected to have lower accuracies on such images as they are devoid of the global object structure. On the other hand, a texture biased CNN may still be able to predict such images with high accuracy. This is because the texture details of the shuffled image patches are still preserved as in the original images. Luo et al. [7] utilized such images to evaluate the shape bias of their proposed CNN architecture. Inspired from this work, Shuffled Image Patches are created for the validation images of ImageNet20 dataset to evaluate our networks.

Shuffled Image Patches are created by splitting an image into the desired number of equal-sized patches and then randomly shuffling those patches before rejoining them. Hence, the position of a patch will be different in the resultant image when compared to the original image. By doing so, the global shape of the object represented in the image will be destroyed. At the same time, since the texture details are local surface statistical cues, they remain preserved in such shuffled patches. The desired number of shuffled patches could be denoted as $n \times n$. For our experiments, the values of n are chosen to be from $\{2, 4, 8\}$. Larger n value denotes that the number of patches are more and hence would result in stronger destruction of the shape information. Figure 5.1 depicts an elephant image along with its corresponding shuffled patches of size 2×2, 4×4, and 8×8.

ImageNet20 contains high-resolution images but they are not of uniform size. Each image in the dataset possesses a different resolution. Hence, it's a general practice in ImageNet

training that the images are first resized to 256×256 and then center cropped to 224×224. The resultant images are of uniform size and these images are used for training a CNN. Similarly, since patch shuffling requires uniform-sized patches and also, to be consistent with the training images, the validation images of ImageNet20 dataset are first resized to 256×256 and then center cropped to 224×224 images which are then split into n × n patches where n $\in \{2, 4, 8\}$.

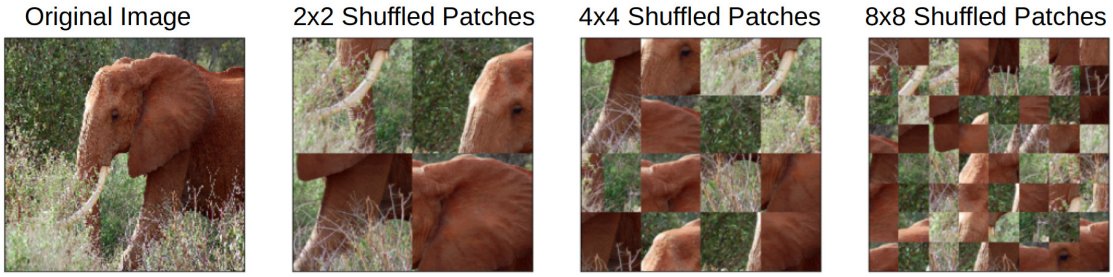

Figure 5.1: An elephant image along with its shuffled patches.

As discussed, a network with strong shape bias is expected to have reduced performance for these Shuffled Image Patches. Also, when a network focuses on global shape rather than the local shape details, a sharp reduction in its accuracy is expected for increased patch sizes. Evaluation results of different networks like the standard network trained on original images, the network trained on Stylized ImageNet (SIN) dataset [1] and our shape biased network trained using edge maps on these Shuffled Image Patches are demonstrated in Chapter 6. It is shown that our proposed network has more shape bias than the network trained on stylized dataset.

5.2 Texture-Shape Cue Conflict Images

Evaluation of CNNs on Shuffled Image Patches gives quantitative result for the shape bias of a network. However, this doesn't provide a direct comparison between the texture and the shape bias exhibited by CNNs. For this purpose, Geirhos et al. [1] proposed a new set of images called Texture-Shape Cue Conflict images. As their name indicates, Texture-Shape Cue Conflict images possess conflicting texture and shape cues and hence, these images include two labels namely a texture label and a shape label. Figure 5.2 represents a few cue conflict images along with their texture and shape labels. For example, the first image of Figure 5.2 contains the shape of a cat while the texture details are from an elephant. When these images are used for evaluation, a direct comparison between the texture and the shape bias of a CNN can be made. When a CNN exhibits more texture bias, its classification would be based on the texture labels. On the other hand, for CNNs with more shape bias, their classification decision will align with the shape labels of the Cue Conflict images.

To create Texture-Shape Cue Conflict images, an iterative style transfer technique proposed

by Gatys et al. [29] is used. For this style transfer, the content images are chosen to be the original RGB images. The style images are considered from the texture dataset of Geirhos et al. [1]. When man-made objects like bottles, clocks, etc., which do not possess unique texture details, are considered for the texture dataset, many such objects are stacked together to produce an image that has its own unique and spatially stationary texture cues. The Cue Conflict images generated by Geirhos et al. [1] are for the ImageNet1000 dataset. Around 16 object categories are utilized for this. Considering 80 images per category, a total of 1280 images are generated.

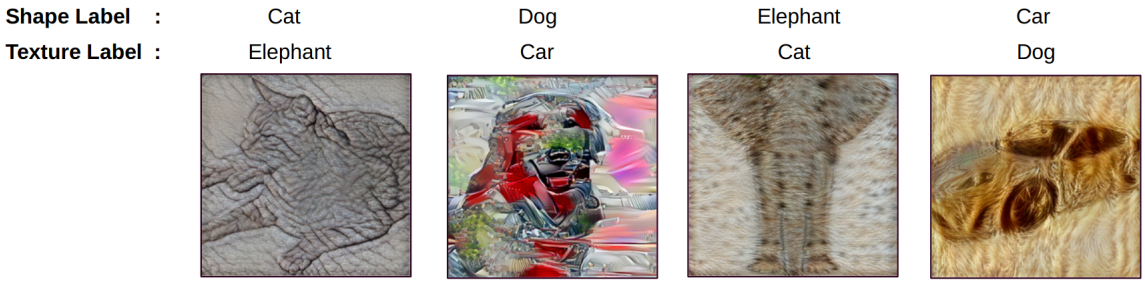

Figure 5.2: Texture-Shape Cue Conflict images.

For ImageNet20 dataset, since not all the Texture-Shape Cue Conflict images generated by Geirhos et al. [1] are relevant, only those images that possess the shape label from one of the 20 classes of ImageNet20 dataset are considered. Among the 16 classes utilized in Texture-Shape Cue Conflict generation, five of them belong to the ImageNet20 dataset. These classes include *Boat, Car, Cat, Dog,* and *Elephant*. There are 400 Cue Conflict images whose shape labels belong to one of these five classes. However, these images could be utilized only for evaluating the number of shape based classifications. For the comparison between the shape and the texture classifications, those images whose texture labels also belong to the ImageNet20 classes are to be considered. Among the 400 Cue Conflict images whose shape label belong to ImageNet20 classes, there exist 100 images whose texture labels also match one of the five above mentioned classes of ImageNet20 dataset. Hence, these 100 Texture-Shape Cue Conflict images are used for the evaluation of texture vs shape bias of a CNN that is trained on ImageNet20 (IN) dataset.

5.3 ImageNet-C Corruptions

For the evaluation of a network's robustness against common corruptions, Hendrycks et al. [12] introduced a benchmark dataset named ImageNet-C that includes different categories of corruptions. CNNs trained on original ImageNet dataset (IN) are shown to be vulnerable to many of these corruptions [12, 14]. Geirhos et al. [1] claimed that improving the shape bias of a network improves its robustness against common corruptions. This hypothesis

is formulated from the fact that a stylized ImageNet trained CNN, with enhanced shape bias, exhibits improved corruption robustness when compared to a standard network trained on natural RGB images. This motivated us to evaluate our shape based network on the corruptions and verify if it shows improved robustness. ImageNet-C corruptions are thus included in our evaluation strategies.

The ImageNet-C dataset comprises of 15 corruptions that belong to the categories noise, blur, weather conditions, and digital transformations. Noise category includes *Gaussian Noise, Shot Noise,* and *Impulse Noise*. Under blur, *Defocus Blur, Glass Blur, Motion Blur,* and *Zoom Blur* are considered. Different weather conditions included are *Snow, Frost,* and *Fog*. *Brightness, Contrast, Elastic Transform, Pixelate,* and *JPEG Compression* are the various types of digital transformations in the ImageNet-C dataset. Figure 5.3 represents an example image of these different distortions.

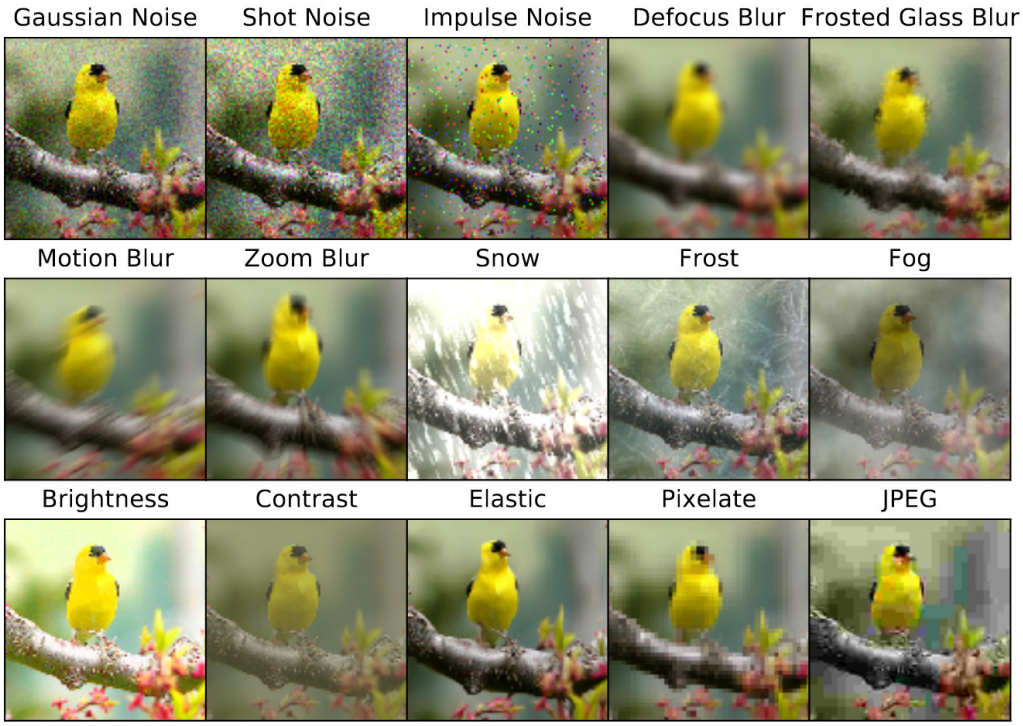

Figure 5.3: Different corruptions included in ImageNet-C dataset [12].

In addition to these 15 distortions, four more distortions are also considered for the validation purpose. Those distortions are *Speckle Noise, Gaussian Blur, Spatter,* and *Saturate*. For each of the ImageNet-C distortions, five levels of severity are considered. Severity level one indicates the least distorted image and the intensity of distortion increases with the increase in severity level. Level five indicates the most corrupted image. Figure 5.4 shows an example image along with its corrupted version on Impulse Noise at different severity levels.

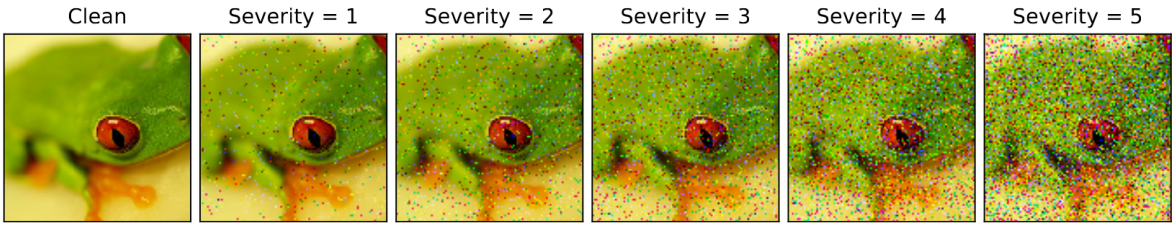

Figure 5.4: An image distorted with Impulse Noise at different severity levels [12].

Different networks considered in this thesis work are evaluated on ImageNet-C distortions and a detailed analysis of the results is presented in Chapter 6.

6 Experiments and Results

This chapter provides details regarding the different training settings considered and also discusses in detail about the results of the conducted experiments along with their analysis. It is organized into the following sections.

1. Experimental Settings - provides the necessary details regarding the experimental setup using which the experiments could be reproduced.
2. Results and Analysis - discusses the results of the considered experiments and provides a detailed analysis of the same.

6.1 Experimental Settings

The experiments considered in this thesis work are carried out using Nvidia GeForce GTX TITAN X GPUs and are conducted on the ImageNet20 dataset using a ResNet18 [25] architecture. Group Normalization [3] along with Weight Standardization [5] is employed in each layer of the ResNet architecture. For the entire set of experiments, the following input transformations from the Torchvision framework are applied before training. *RandomResizedCrop* with a crop value of 224 and the scale range between (0.2, 1.0) is used. This acts like a data augmentation method that crops an image with a random size range between 0.2 and 1 of the original size with a random aspect ratio between 0.75 and 1.33 of the original value and then finally resizes it to a 224×224 image. An additional data augmentation named *RandomHorizontalFlip* that uses a random probability of 0.5 to make a horizontal flip on the input is also utilized. The resulting images are then transformed to Tensors using *TransformToTensor* augmentation and are fed to the network for training. The training batch size is fixed as 128. A Stochastic Gradient Descent (SGD) [64] optimizer with a momentum of 0.9 and a weight decay rate of 10^{-4} is used. Since it's an image classification task, the loss function used for our network is the Cross Entropy loss.

To avoid complexities, the networks trained using different datasets are referred by the name of their datasets itself. To be specific, a CNN trained on standard ImageNet20 (IN) dataset is referred as *IN* and a network trained on Stylized ImageNet (SIN) is identified as *SIN*. Further, our shape based CNN that is trained using the Edge Dataset is termed as *E*. On the other hand, the network that is trained only on the edge maps is referred as *EdgeCNN*. Different EdgeCNNs that are trained using the different variants of edge maps are referred by the name of those variants. In addition to these, CNNs are trained on different style augmentation datasets namely Stylized Edges, Intra-Stylized ImageNet, Intra-Stylized Edges,

and Superposition and those networks are referred as *SE, I-SIN, I-SE* and *SE+IN* respectively. Further, although the networks *E* and *SIN* are both aimed at enhancing the shape bias, they have different perspectives. The network *E* directly focuses on improving the shape bias while *SIN* reduces the texture dependency of the network which in turn improves its shape bias. This shows that these two methods complement each other. Hence, an additional setting termed Hybrid CNN is considered that utilizes both Edge maps and SIN dataset in its training. Such Hybrid networks are termed as *E-SIN*.

The network (*IN*) and the variants of *EdgeCNN* are trained under the following setting. They are trained for 100 epochs with an initial learning rate of 0.1. Later, the learning rate is reduced to 0.01 at 60th epoch and is further decreased to 0.001 at 90th epoch. All other considered networks except *E-SIN* had undergone a two-staged training namely, the initial training and finetuning. During the initial training, the networks are trained only on their respective datasets for 75 epochs with a learning rate of 0.1 for the first 60 epochs and 0.01 for the last 15 epochs. During finetuning, additionally the standard ImageNet (IN) data is also fed to the networks. This finetuning is carried out for another 75 epochs with an initial learning rate of 0.01 that is reduced to 0.001 at 60th epoch. An equal number of samples from both the datasets are fed to the network during each epoch of the finetuning. For example, the network *E* receives 128 images from the ImageNet20 (IN) dataset and another 128 images from the Edge Dataset (E) during its finetuning.

For the network *E-SIN*, its training is again two-staged with a slight modification to the above setting. During the initial training of *E-SIN*, the Edge Dataset (E) is used. However, in the finetuning stage, the Stylized ImageNet (SIN) dataset along with the ImageNet20 (IN) data is used. This implies that E-SIN has seen three different datasets in its whole training process namely *E, SIN* and *IN* while all other networks are just exposed to two datasets.

Among the two-staged training settings, the networks that involve edge maps namely *E, SE* and *I-SE* are finetuned using a slightly different loss function when compared to others. During finetuning, the rest of the networks use a cumulative loss function from both the datasets. However, since the distribution of edge maps vary a lot than the Standard RGB images (IN), finetuning with equal loss weightage would result in a reduced validation accuracy. Hence, to compensate this, the Loss function (L) considered during the finetuning of *E, SE* and *I-SE* uses an additional loss weightage term λ between the two datasets. This function is defined as follows:

$$\begin{aligned} L := L_{\text{IN}} \; + \; & \lambda \cdot L_{\text{Edgemaps}} \\ & where, \;\; \lambda = 0.01 \\ & \qquad L_{\text{Edgemaps}} \in \{L_{\text{E}}, L_{\text{SE}}, L_{\text{I-SE}}\} \end{aligned} \tag{6.1}$$

For all the networks with two-staged training, the parameters of the first convolutional layer and the affine parameters of the first normalization layer are frozen during finetuning. This

is because it has been observed that, rather than finetuning all the parameters of a network, freezing its initial layers results in a stronger shape bias. Further, since the networks use standard ImageNet (IN) dataset for finetuning, they can still learn the local texture statistics. To avoid this, *style randomization* method introduced in Chapter 4, which reduces the possible texture bias of the CNNs, is used during finetuning. ResNet18 architecture includes a series of eight residual blocks split into four layers and *style randomization* is employed at the beginning of each layer for the natural RGB images. Compared to the other settings like no stylization or *style blending* [8], *style randomization* encodes more shape bias and these results are discussed in the following section.

6.2 Results and Analysis

6.2.1 *IN* vs *EdgeCNN*

This subsection provides a detailed analysis of the evaluation between *IN* and *EdgeCNN* to verify if the usage of edge maps would help in improving the shape bias of a network. Here, Binary Edges are used for training the *EdgeCNN*. Since the distribution of edge maps is quite different than that of the original RGB images, *EdgeCNN* cannot be validated on original IN dataset and it may result in poor performance.

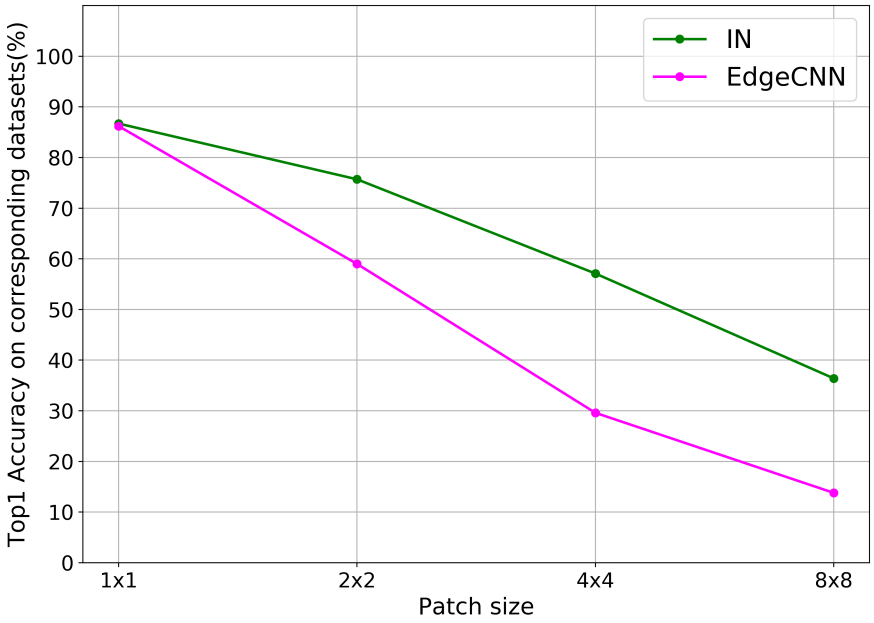

Figure 6.1: Evaluation results of IN and EdgeCNN on corresponding shuffled image patches. Lower accuracy indicates higher shape bias.

Hence, the results shown in Figure 6.1 refer to the performance of the networks *IN* and *EdgeCNN* on the shuffled image patches of their corresponding datasets. To be precise, the evaluation of *EdgeCNN* is carried out on the edge maps (E) while the network *IN* is evaluated on the original images (IN). For a fair comparison, the models are chosen such that the validation accuracy of *IN* matches with that of *EdgeCNN*. It is evident from Figure 6.1 that, *IN* has no sudden drop in validation accuracy when evaluated on shuffled patches. This is due to its strong texture bias. Even for 8×8 patches, the accuracy remains around 37%. On the other hand, for *EdgeCNN* an abrupt drop in the validation accuracy is observed even for 2×2 image patches. This shows that the network focuses more on the global shape details. It could be also noted that there is an accuracy difference of more than 25% between *IN* and *EdgeCNN* for 4×4 patches. This provides evidence that the lack of texture information in Edge maps forced the network to learn global shape details. Hence, Edge maps could be used for shape based learning.

6.2.2 Variants of *EdgeCNN*

Different *EdgeCNNs* are trained using the edge map variants discussed in Chapter 4 and their results are analyzed to select one final variant for the training of our shape based network. Figure 6.2 represents the evaluation results of the different variants of *EdgeCNNs* on the corresponding shuffled image patches.

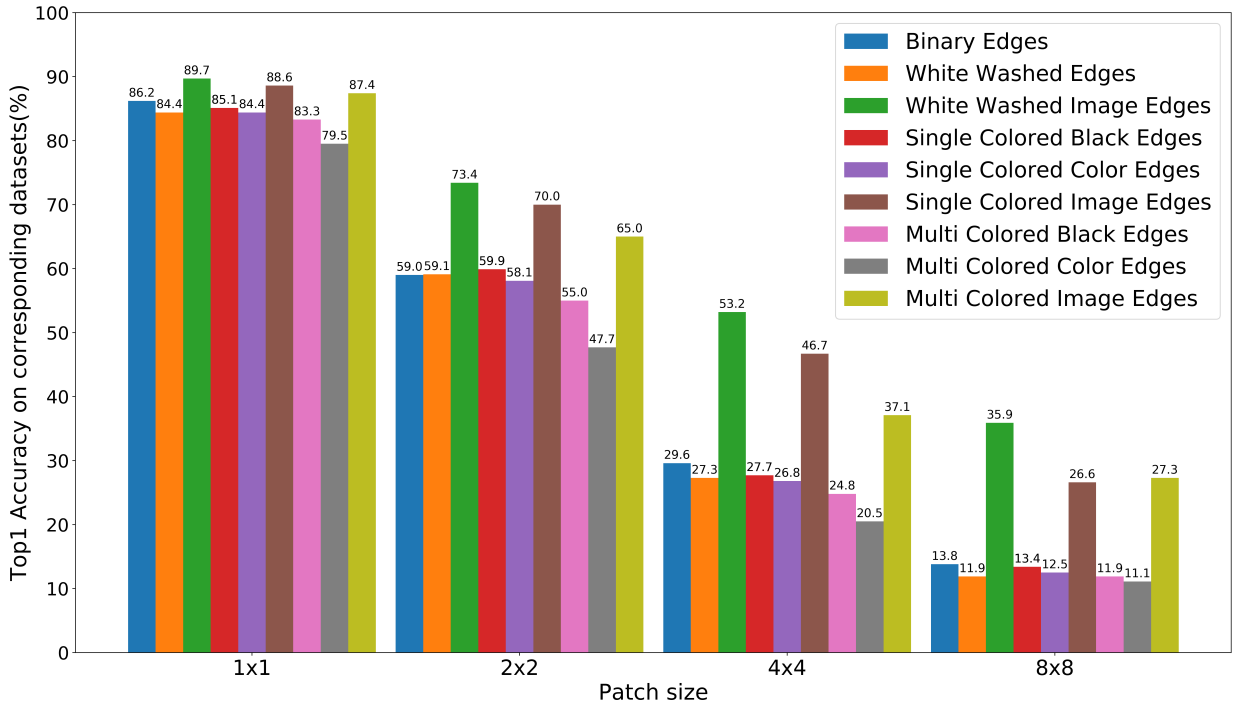

Figure 6.2: Evaluation results of different *EdgeCNN* variants on corresponding shuffled image patches. Lower accuracy indicates higher shape bias.

The results shown in Figure 6.2 provide the following inferences. The edge variants *White Washed Image Edges, Single Colored Image Edges* and *Multi Colored Image Edges* have higher validation accuracies among others. However, they have higher accuracies even on the shuffled image patches in which the global shape details are destroyed. These results indicate that these edge variants still utilizes some texture details that are preserved in the image edges and are used for their classification. Therefore, utilizing such datasets for shape learning may not be a wise choice. Among the other variants, edge maps with multi-colored background namely *Multi Colored Black Edges* and *Multi Colored Color Edges* have lower validation accuracies than others. Since validation accuracy is an equally important factor for the choice of edge maps, these variants can also be ignored. The performance results of the remaining variants *Binary Edges, White Washed Edges, Single Colored Black Edges* and *Single Colored Color Edges* are approximately same. Hence, the validation results of these *EdgeCNN* variants on ImageNet20 validation dataset (IN) are analyzed. Figure 6.3 represents the validation as well as the patch shuffled results of different *EdgeCNN* variants on IN dataset.

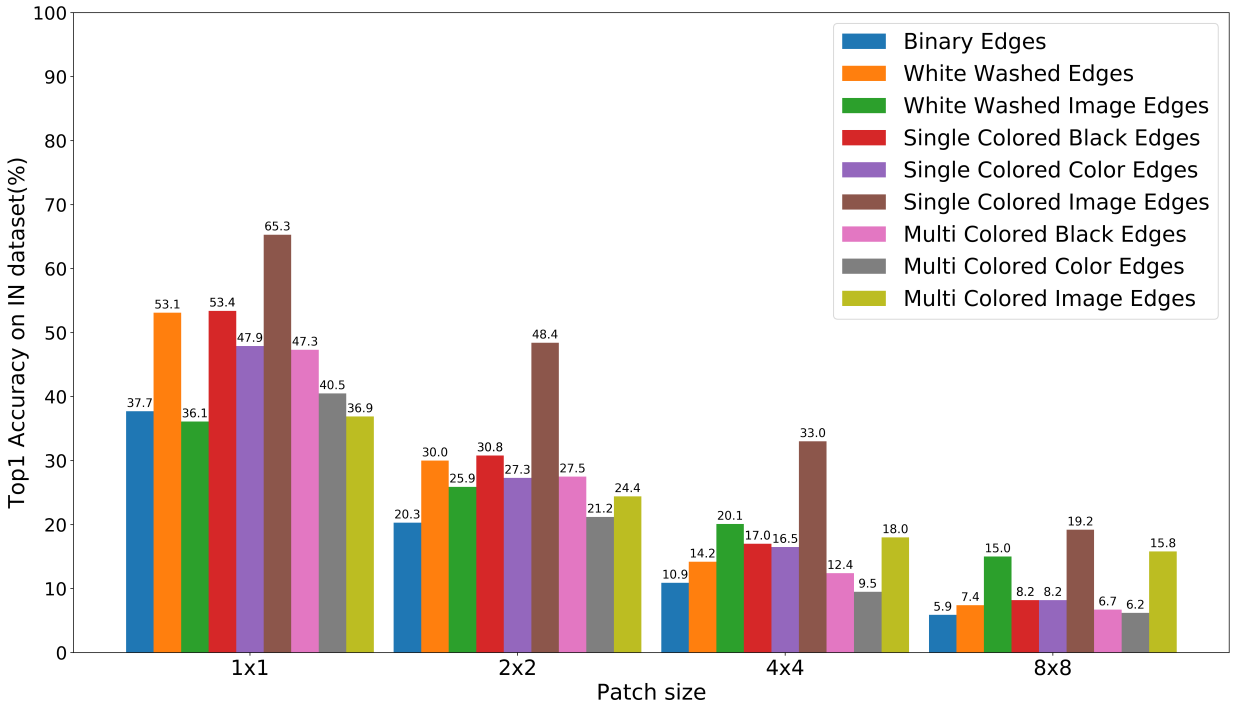

Figure 6.3: shuffled patch results for different EdgeCNN variants on IN dataset. Lower accuracy indicates higher shape bias.

It is evident from Figure 6.3 that the validation accuracy of *Binary Edges* on IN dataset is lower than that of the *White Washed Edges, Single Colored Black Edges* and *Single Colored Color Edges* and hence it can be ignored. *Single Colored Color Edges* also has lower validation accuracy compared to the other two settings. So, the final filtered settings are *White Washed Edges* and *Single Colored Black Edges*. Comparing the shuffled patch accuracies of these

two settings, particularly the 4×4 patches, *White Washed Edges* seem to be better than *Single Colored Black Edges*. Further, since the background color for *Single Colored Black Edges* will be randomly chosen every time, their results may not be consistent across each evaluation though that may cause only a minor variance. Hence, the *White Washed Edges*, which has slightly better performance than *Single Colored Black Edges* and which would provide consistent results across many evaluations, is the final choice. Henceforth *White Washed Edges* are used for training those networks that involve edge maps including our shape based network *E*.

6.2.3 Evaluation of the Shape based Network (*E*)

Results on Shuffled Image Patches

For evaluating our shape network *E*, the networks *IN* and *SIN* are considered as the baselines. Figure 6.4 provides the evaluation results of these three networks on shuffled patches of ImageNet20 (IN) validation data. It can be seen that *IN* has higher accuracies on the shuffled image patches when compared to *SIN* and *E*. Further, though the accuracies of *E* and *SIN* are not much different for 2×2 and 4×4 patches, our shape network *E* has lower accuracy than *SIN* for 8×8 shuffled patches where the global shape details are completely destroyed. This implies that *SIN* might have focused on some local shape details while *E* did not.

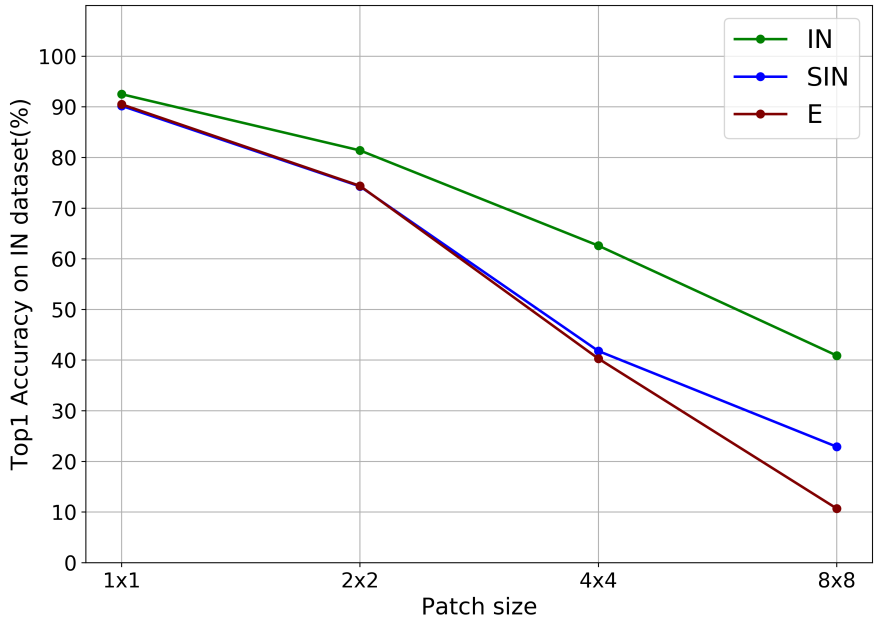

Figure 6.4: Evaluation results of *IN, SIN* and *E* on shuffled image patches of IN dataset. Lower accuracy indicates higher shape bias.

Further, in Figure 6.4, the validation accuracy of *IN* seems to be a bit higher than *SIN* and *E*. To get clear insights if this difference in validation accuracies has any impact on the accuracies of shuffled patches, another evaluation is carried out where only those validation images that are correctly classified by all the three networks *IN, SIN, E* are accounted. Figure 6.5 shows the result of such an evaluation. Among the 1000 validation images of ImageNet20 dataset, 791 validation images are correctly classified by all the three networks and those 791 images are considered here for the evaluation. It can be seen from Figure 6.5 that the validation accuracy of all three networks is 100%. Further, the results show that there is a clear deviation in the shuffled patch accuracies of *IN* when compared to *SIN* and *E*. Also, quite interestingly, their accuracy difference increases with the increase in patch size. The results indicate that the shape based networks *SIN* and *E* suffer a larger drop in accuracy when the global shape details are more destroyed. Also, as already discussed, the result on 8×8 patches shows that *E* exhibits more global shape bias than *SIN*.

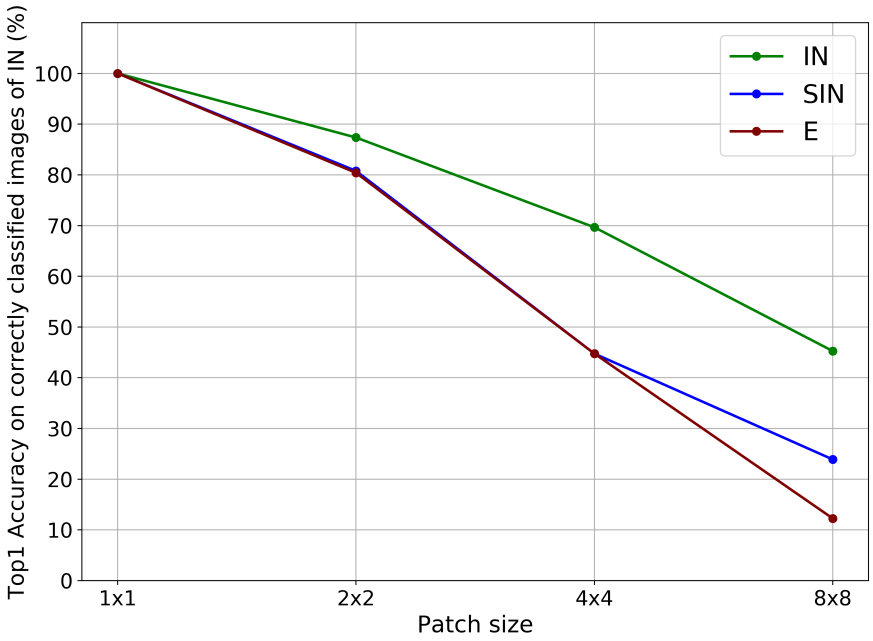

Figure 6.5: Shuffled patch results of *IN, SIN* and *E* on the correctly classified images of IN dataset. Lower accuracy indicates higher shape bias.

Results on Texture-Shape Cue Conflict Images

In addition to the shuffled patches evaluation, the networks *IN, SIN* and *E* are evaluated on the texture-shape cue conflict images discussed in Chapter 5 and their results are presented in Table 6.1. The second column in the table represents the shape based results among the 400 cue conflict images whose shape labels correspond to one of the classes of ImageNet20

dataset (IN). Further, the third and the fourth column of Table 6.1 represent the shape and texture based results for 100 cue conflict images whose texture as well as shape labels belong to ImageNet20 classes.

Network	#400 Cue Conflict Images	#100 Cue Conflict Images	
	Shape results	***Shape results***	***Texture results***
IN	63	11	39
SIN	143	28	5
E	151	28	24

Table 6.1: Comparison of texture-shape cue conflict results between *IN, SIN* and *E*.

The evaluation results of *IN* indicate that the network has high texture bias.. On the other hand, the networks *SIN* and *E* exhibit stronger shape bias than *IN*. At the same time, the texture results are low for *SIN* that is trained on stylized dataset. This is because the texture details are meaningless for such stylized images. When compared to *SIN*, *E* has higher number of texture based results. This can be owed to the fact that, during finetuning, our shape based network *E* might have had the freedom to explore the texture information available in the original IN data. Nevertheless, the shape results show that *E* exhibits strong shape bias as similar to *SIN*.

6.2.4 Effects of Style Randomization

Results on Shuffled Image Patches

Style randomization that is discussed in Chapter 4 is introduced in this thesis work to reduce the texture dependency of networks by randomizing the statistics of feature maps. A detailed evaluation of this method by comparing it against a network without any randomization and a network with style blending is presented in this subsection. Three network variants, *IN, SIN* and *E* are considered for this evaluation. The first setting is without any stylization, the second one has *style blending* and the third setting includes *style randomization*. Table 6.2 lists the validation results on 4×4 shuffled patches for all the three networks under these three settings. This counts to a total of nine different network settings. Further, since style blending and style randomization makes the classification task difficult, the validation accuracies of these nine networks are quite different. To have a meaningful comparison, the data provided in Table 6.2 includes the evaluation results of only 598 validation images of IN dataset that are correctly classified by these nine different variants. It can be seen that for the three networks *IN, E,* and *SIN*, the patch shuffled results of *style blending* is lower than that of the setting without any stylization. However, *style blending* has higher accuracies than *style randomization* for all the networks. Further, among the nine settings, the network *E* with *style randomization* has the lowest 4×4 patch shuffled accuracy. These results imply

that randomizing the statistics of feature maps improves the shape bias of the networks. Further, when the statistics are chosen from out-of-distribution data, the networks show better results.

Network	4×4 Shuffled Image Patches		
	No Stylization	***Style Blending***	***Style Randomization***
IN	67.22	51.34	41.97
SIN	38.46	36.96	34.95
E	34.11	33.95	**28.43**

Table 6.2: Comparison of different feature space style augmentation methods on 4×4 shuffled image patches. Lower accuracy indicates higher shape bias.

Results on Texture-Shape Cue Conflict Images

The effect of stylizations on the three networks *IN*, *SIN* and *E*, are also evaluated using texture-shape cue conflict images. For this purpose, the 400 cue conflict images whose shape label corresponds to one of the 20 classes of ImageNet20 dataset (IN) are considered. When a network has more shape bias, it should be able to classify most of the cue conflict images according to their shape labels. It can be seen from Table 6.3 that for each of the network setting, *style randomization* has better results. Also, similar to the patch shuffled results, the shape network *E* proposed in this thesis work has the highest shape based results in all three scenarios. Considering these evaluation results, *style randomization* is utilized for each of the network variants included in this thesis work and their results are discussed in the next subsection.

Network	Shape based results among 400 Cue Conflict Images		
	No Stylization	***Style Blending***	***Style Randomization***
IN	68	82	86
SIN	144	155	156
E	155	166	**193**

Table 6.3: Comparison between different feature space style augmentation methods for the shape based results of cue conflict images.

6.2.5 Evaluation of *IN, SIN, E, SE & E-SIN* with Style Randomization

Results on Shuffled Image Patches

Figure 6.6 represents the shuffled patch accuracies for the following networks with style randomization: Standard CNN (*IN*), Stylized CNN (*SIN*), our shape based network (*E*), Stylized Edge CNN (*SE*), and Hybrid CNN (*E-SIN*). The models are chosen such that all these networks have almost similar validation accuracies. It has to be noted that a higher reduction in accuracy on shuffled patches implies a stronger shape bias. Also, a network that depends more on the global object shape is expected to have a larger drop in accuracy when the patch size increases. It can be seen from the results of *IN* that style randomization helped in reducing the texture bias of standard network to a great extent. However, other shape based networks exhibit stronger shape bias than *IN*. It can be also inferred that the global shape bias is prominent for *E* when compared to *SIN* [1]. In addition, *SE* exhibits better shape bias than *E* while *E-SIN* seems to have the highest shape bias among the considered networks. These results imply that the style variations in *SE* helped with more shape learning than simple White Washed Edges. Also, since *E-SIN* is trained on two complementing datasets *E* and *SIN*, it shows stronger shape bias than other networks.

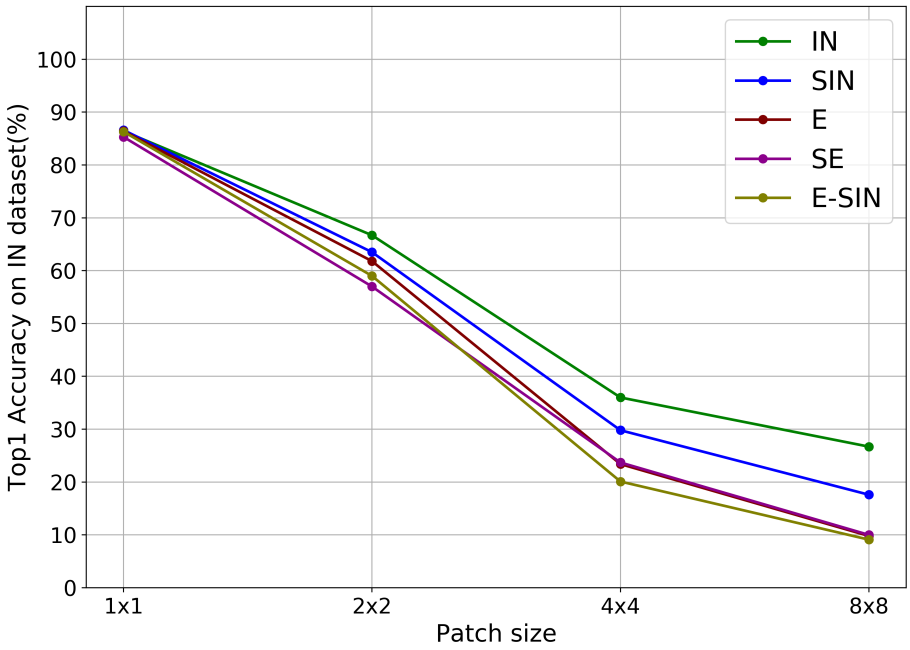

Figure 6.6: Shuffled patch accuracies of the networks *IN, SIN, E, SE,* and *E-SIN* with style randomization on the entire validation dataset of IN. Lower accuracy indicates higher shape bias.

In addition to the above evaluation, Figure 6.7 includes the shuffled patch accuracies of the aforementioned networks only on those images that are correctly classified by each of them. Among the 1000 validation images of ImageNet20 dataset 644 images are correctly classified by all these networks. Figure 6.7 represents the evaluation results of these networks on those 644 images. It can be seen that this plot follows the same pattern as in Figure 6.6. Global shape details encoded in the networks increase in the following the order: *IN* < *SIN* < *E* < *SE* < *E-SIN*.

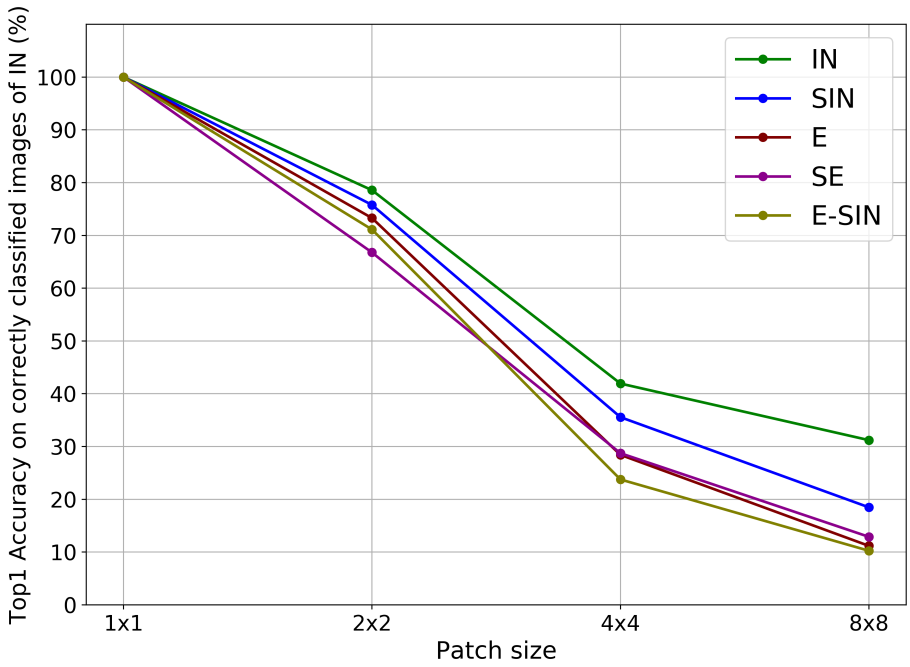

Figure 6.7: Shuffled patches accuracies of the networks *IN, SIN, E, SE,* and *E-SIN* with style randomization on the correctly classified validation images of IN dataset. Lower accuracy indicates higher shape bias.

Results on Texture-Shape Cue Conflict Images

Table 6.4 contains the shape based results of *IN, SIN, E, SE* and *E-SIN* for five different classes of the texture-shape cue conflict images. These classes include Boat, Dog, Cat, Elephant, and Car. There are 80 images per class which totals to 400 cue conflict images. The accumulated results on these 400 cue conflict images are also listed in the last column of Table 6.4. These results seem to be in par with the shuffled patch results. The standard network *IN* exhibits the lowest shape bias with 86 correct classifications. It has least result on Dog images with only three correct classifications. All other networks exhibit higher shape bias than *IN*. Though *SIN* could classify 156 images based on their shape, it still finds hard to classify the Dog images while the other networks have better results on them. Our shape

based network using edge maps *E* classified 193 out of 400 images based on their shape labels. As similar to shuffled patch results, *SE* has more shape bias than *E* and the network *E-SIN* encodes the strongest shape bias among others.

	Shape based results among 400 Cue Conflict Images					
Network	***Boat***	***Dog***	***Cat***	***Elephant***	***Car***	***Total***
IN	11/80	3/80	22/80	13/80	37/80	86/400
SIN	21/80	3/80	44/80	28/80	60/80	156/400
E	34/80	19/80	47/80	41/80	52/80	193/400
SE	27/80	15/80	62/80	56/80	64/80	224/400
E-SIN	38/80	20/80	68/80	43/80	65/80	**234/400**

Table 6.4: Comparison between different networks *IN, SIN, E, SE,* and *E-SIN* with style randomization for the shape based classification of 400 cue conflict images.

Comparison between the texture and the shape bias of a CNN can be carried using those cue conflict images whose texture as well as shape labels belong to the ImageNet20 classes. Among the 400 cue conflict images considered for the previous shape based evaluation, 100 images have their texture labels also within the classes of ImageNet20. Table 6.5 thus shows the comparison between the texture and the shape classifications of the networks *IN, SIN, E, SE* and *E-SIN* on these 100 images. This table also contains the classification results for each class separately. It can be seen that *IN* has the lowest shape based results while at the same time, it has the highest texture based classifications. This explains the texture bias of *IN*. Further, it has to be noted that *style randomization* helped in reducing the texture bias of *IN* and so its shape and texture results seem comparable. Without style randomization, number of texture classifications are far higher than the shape results of *IN*.

	Shape based results						**Texture based results**					
Network	***Boat***	***Dog***	***Cat***	***Elephant***	***Car***	***Total***	***Boat***	***Dog***	***Cat***	***Elephant***	***Car***	***Total***
IN	1/20	1/20	9/20	2/20	5/20	18/100	4/20	5/20	6/20	4/20	1/20	20/100
SIN	4/20	0/20	11/20	6/20	11/20	32/100	1/20	0/20	1/20	0/20	0/20	**2/100**
E	2/20	7/20	14/20	11/20	12/20	46/100	5/20	5/20	2/20	1/20	2/20	15/100
SE	4/20	5/20	16/20	14/20	16/20	55/100	2/20	1/20	1/20	2/20	0/20	6/100
E-SIN	6/20	6/20	19/20	11/20	16/20	**58/100**	2/20	3/20	0/20	1/20	0/20	6/100

Table 6.5: Comparison of texture and shape results between different networks *IN, SIN, E, SE,* and *E-SIN* with style randomization on 100 cue conflict images.

Among the other considered networks, *E-SIN* that is trained using both edge maps and the SIN dataset has the highest shape results. It also has low texture based classifications. Our shape network *E* has higher shape results than *SIN* while it also has more texture results than *SIN*. This can be related to the fact that SIN dataset has irrelevant texture details which would have forced the network not to focus on such cues even during its finetuning. Further, since the distribution of SIN is similar to that of the IN dataset, no loss weightage is required during its finetuning. On the other hand, owing to the distribution of edge maps, loss weightage is used for the finetuning of network *E* . This would have allowed the network to learn texture details too. *Style randomization* helped in reducing this texture bias but not to a great extent. Nevertheless, the shape based results indicate that the network *E* has more global shape bias than *SIN*.

Comparing the results of both the cue conflict evaluations, the shape bias of *IN* seems to be the lowest. This is followed by *SIN* that has improved shape bias with the lowest texture bias. The network *E* encodes stronger shape bias than *IN* and *SIN* though it exhibits a little higher texture bias. The network *SE* trained on stylized edge maps has better shape results than the above three networks while the hybrid network *E-SIN* has the strongest shape bias among others owing to its advantage of being trained on both SIN and E datasets.

6.2.6 Evaluation of *IN, SIN & E* on ImageNet-C distortions

Geirhos et al. [1] showed that CNNs trained on stylized ImageNet (SIN) have improved robustness on ImageNet-C corruptions when compared to a standard network. Further, the improved shape bias of such networks is presumed to be the reason behind their improved corruption accuracies. Since our shape based network (*E*) also exhibits a stronger shape bias, it is also evaluated on the ImageNet-C distortions to verify if it shows similar robustness improvement towards corruptions. For this evaluation, other than *E* and *SIN*, two variants of *IN* are considered as the baselines. The first one is the general baseline network without any stylization and the other variant includes *style randomization* in it. For all these networks, the models with the best validation accuracies are chosen for the evaluation.

Network	***Noise(%)***	***Blur(%)***	***Weather Conditions(%)***	***Digital Transforms(%)***	***mCA(%)***
IN (no stylization)	50.4	57.81	69.25	76.03	64.69
IN (style randomization)	59.57	62.7	72.4	78.85	69.39
SIN	74.54	72.67	78.36	83.05	**77.64**
E	60.65	48.07	66.09	71.54	62.01

Table 6.6: Comparison of the validation accuracies of *IN, SIN* & *E* on different categories of ImageNet-C distortions.

As discussed in Chapter 5, ImageNet-C dataset contains 15 different distortions under four categories namely Noise, Blur, Weather Conditions and Digital Transformations. Further, each of these distortions includes five different severity levels. Mean validation accuracies for each category of distortions are presented in Table 6.6. These accuracies are calculated by considering the mean accuracies of all the distortions under a particular category across all five severity levels. Table 6.6 contains these results on *IN* without stylization, *IN* with style randomization, *SIN*, and *E*. The last column in the table represents the mean Corruption Accuracies (mCA) of these networks that is averaged over all the 15 distortions across the five severity levels.

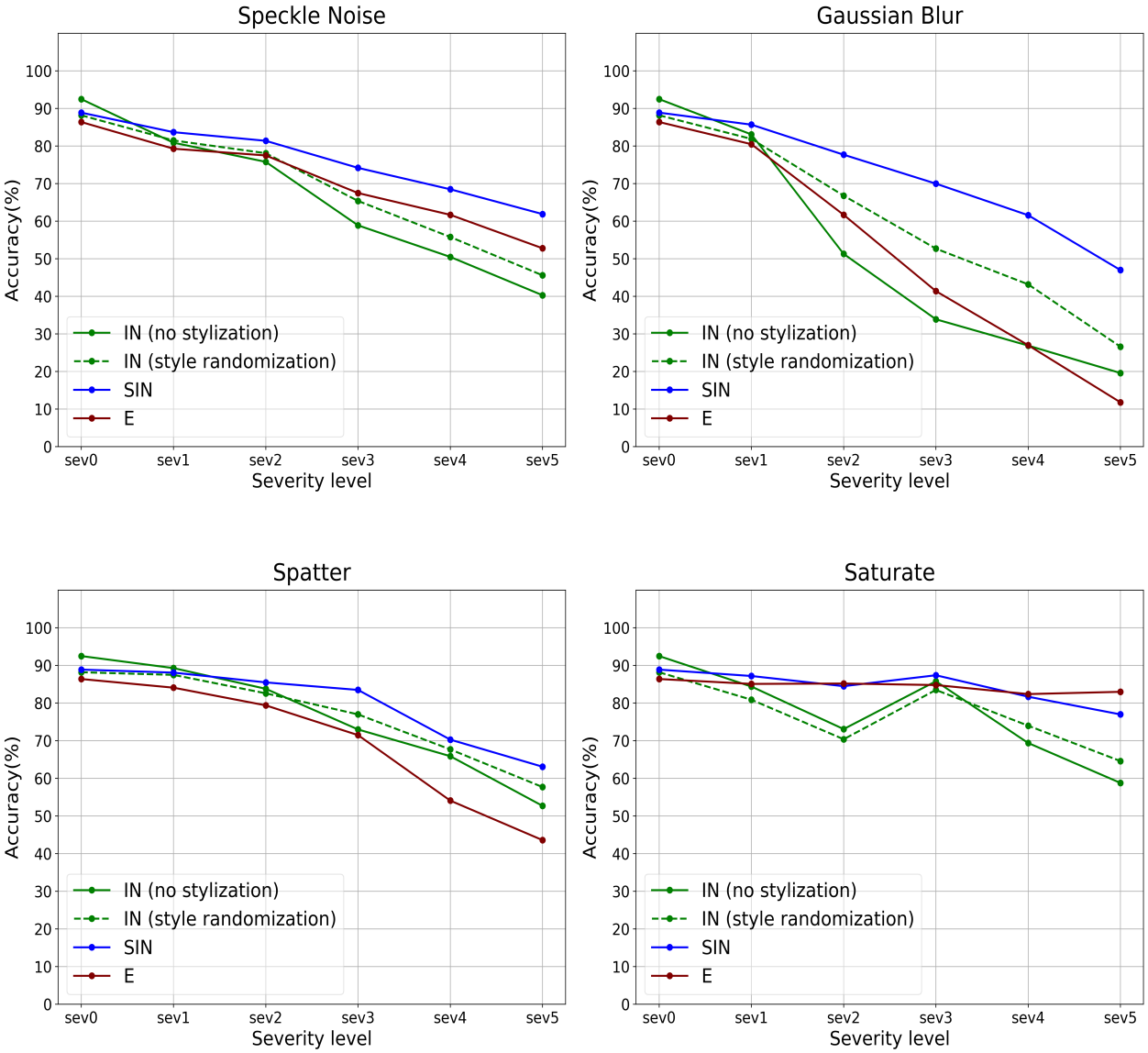

Figure 6.8: Performance of *IN*, *SIN* & *E* on the validation distortions of ImageNet-C at different severity levels.

In addition to Table 6.6, the plots in Figure 6.8 also represent the distortion accuracies of the considered networks on Speckle Noise, Gaussian Blur, Spatter and Saturate for different severity levels. These four are the validation distortions listed in ImageNet-C dataset [12] and each of them belongs to a different distortion category. The results presented in Table 6.6 indicate that the corruption performance for *IN* with style randomization is improved across each category when compared to the network *IN* without any stylization. This difference can also be visualized with the validation distortions represented in Figure 6.8. Further, it can be inferred from each plot of Figure 6.8 that the *SIN* network with more shape bias has improved distortion accuracies when compared to both the variants of *IN*. This also stands true for the 15 distortions whose averaged accuracies are listed in Table 6.6. The corruption results between the two *IN* variants and also the results of *SIN* indicate that the robustness against corruptions increases with an increase in the shape bias of a network.

However, network *E*, which also has high shape bias as similar to *SIN*, seems to have poor performance in most of the corruption categories when compared to both the variants of *IN*. On comparing the results between *IN* (no stylization) and *E*, the shape-biased network *E* shows improved performance only for the noise category. The same can also be observed from Figure 6.8. Nevertheless, when the network *E* is compared against *IN* that includes style randomization, there is no substantial difference even for the noise category. On other categories, the shape network *E* has poor results when compared to both the variants of *IN*. In particular, for distortions in the blur category, the mean distortion accuracy is much lower for *E*. It can also be inferred from Figure 6.8 that the validation accuracy of the network *E* on Gaussian Blur drops rapidly with the increase in the severity level than any other network. The mean accuracy across all distortions is also the lowest for our shape based network *E* and has an approximate difference of 7% when compared to the standard *IN* network with style randomization. Comparing the results of *E* with *SIN*, there is a clear difference in their distortion accuracies. Results from Table 6.6 imply that there exists a minimum accuracy difference of 10% for each of the categories between these two networks. Among all other categories, blur has the highest deviation of around 24%. The mean Corruption Accuracy (mCA) of *SIN* is 77.64% while for *E*, it is only 62.01%.

These evaluation results on ImageNet-C distortions clearly show that two different networks *SIN* and *E*, both with improved shape bias, have wide variations in their corruption robustness. While the *SIN* network shows improved robustness than the texture-biased network (*IN*), the other network *E* has poor results when compared to the standard network *IN*. This stood as the motivation for us to conduct further experiments in understanding the actual role of shape bias in improving the corruption robustness of CNNs.

6.2.7 Influence of Shape Bias on Common Corruptions

Different networks including the style variants that are discussed in Chapter 4 are evaluated on ImageNet-C corruptions. This may give a detailed insight into the factors contributing to the improved corruption robustness of the network *SIN* that is trained on the stylized dataset. In addition to the previously considered networks (*IN, SIN and E*), the other networks included

are Stylized Edge CNN (*SE*), Intra-Stylized Edge CNN (*I-SE*), Intra-Stylized CNN (*I-SIN*), Hybrid CNN(*E-SIN*) and Superposition with α=0.5 (*SE+IN*). Among the two variants of standard CNN (*IN*), the network without any stylization is considered to be the base model here. All other considered networks are incorporated with style randomization in their architecture. The best models of each network are chosen for the distortion evaluation.

Network	***Noise(%)***	***Blur(%)***	***Weather Conditions(%)***	***Digital Transforms(%)***	***mCA(%)***
IN	50.4	57.81	69.25	76.03	64.69
SIN	74.54	72.67	78.36	83.05	77.64
E	60.65	48.07	66.09	71.54	62.01
SE	69.32	64.64	73.51	78.0	71.81
I-SE	64.73	64.38	72.49	77.05	70.3
I-SIN	75.19	73.19	78.6	82.93	77.92
E-SIN	72.03	58.31	73.51	80.67	71.55
SE+IN	75.05	75.62	79.47	83.66	**78.96**

Table 6.7: Comparison of the validation accuracies of various networks on different categories of ImageNet-C distortions.

Table 6.7 could be considered as an extension of Table 6.6 where in addition to the category-wise mean corruption accuracies of *IN, SIN* and *E*, the results of *SE, I-SE, I-SIN, E-SIN* and *SE+IN* are also listed. Among these eight networks, *E* has the least mCA though it has high shape bias. The network *SE* that is trained on stylized edge maps has improved accuracy than the standard network *IN*. This shows that stylization has some role in improving corruption robustness.

Further, the results of the network *I-SE* seem comparable to the network *SE* except for the noise category. It has to be noted that the only difference between the two networks *SE* and *I-SE* is that the styles used for the style transfer come from out-of-distribution data for the former, while they are taken within the ImageNet20 (IN) dataset itself for the latter. The mean Corruption Accuracy (mCA) of *SE* is slightly higher than that of *I-SE*. Similar to the above two networks, the difference between Stylized-ImageNet (*SIN*) and Intra-Stylized ImageNet (*I-SIN*) is also just the distribution of their styles. The corruption accuracies of *SIN* and *I-SIN* for different distortion categories also remain quite similar. Here, mCA of *I-SIN* is little higher than that of *SIN* results. Such a comparison of distortion results between *SE* & *I-SE* and also between *SIN* & *I-SIN* indicates that the style images used for style transfer need not necessarily be considered from out-of-distribution data to have improved corruption robustness.

The distortion performance of Hybrid network (*E-SIN*) also doesn't seem to be as good as Stylized network (*SIN*) though it has the strongest shape bias. Despite the fact that *E-SIN* is also finetuned using the stylized dataset, its initial training is however done using the edge maps. Further, as explained earlier, during the finetuning of *E-SIN* on stylized images, its initial layers are frozen. Hence, the network *E-SIN* may not have seen enough of stylized data when compared to *SIN* so that they have similar corruption robustness. Figure 6.9 contains the plots for all these networks on the validation distortions of the ImageNet-C dataset.

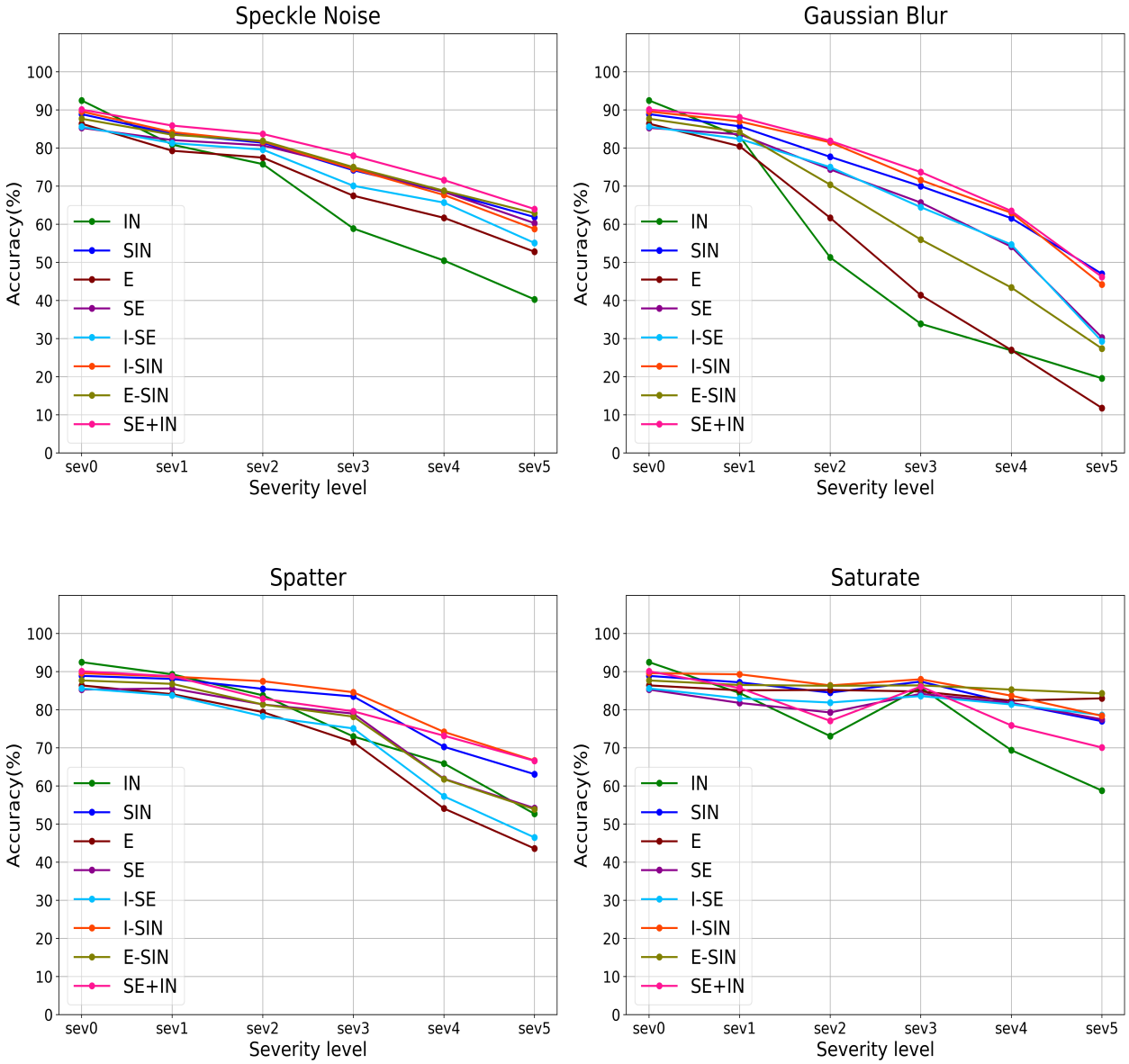

Figure 6.9: Performance of various networks on the validation distortions of ImageNet-C at different severity levels.

Among all the considered CNNs, the only network that has slightly improved corruption robustness than *SIN* is the Superposition network (*SE+IN*). It has to be remembered that the network *SE+IN* is trained and finetuned using a different set of images that are generated through the interpolation between stylized edges and the original ImageNet dataset. Further analysis of the possible reasons behind the improved distortion accuracies of *SE+IN* is explained with the help of Table 6.8. This table includes detailed information regarding the input image compositions for the networks *IN, SIN, E, SE* and *SE+IN*. Besides, it also carries the number of shape vs texture predictions of these networks among 100 cue conflict images. Such an evaluation gives a clear insight regarding the shape bias of the networks. The last column represents their mean accuracies on ImageNet-C distortions.

	Input Image Composition			**#100 Cue Conflict Images**		
Network	***Natural image***	***Edge map***	***Style transfer***	***Shape results***	***Texture results***	***mCA(%)***
IN	✓	✗	✗	11	39	64.69
SIN	✓	✗	✓	34	2	77.64
E	✗	✓	✗	46	15	62.01
SE	✗	✓	✓	55	6	71.81
SE+IN	✓	✓	✓	22	13	**78.96**

Table 6.8: Analysis of different networks using their input compositions, texture/shape accuracy on cue conflict images, and Mean corruption accuracy.

The factors that constitute the input images of these networks include some preserved characteristics of the natural images, explicit edge maps, and stylization. It can be seen from Table 6.8 that the input images for the network *IN*, which are the natural images with preserved statistics don't include any explicit edge maps and also have not undergone any stylization. *IN* trained using such images possess high texture bias and have low corruption robustness. This implies that the natural image statistics alone are not sufficient for a network to gain robustness against corruptions. The input to the network *SIN* is the stylized images that are generated through stylization. It has to be also observed that these stylized images possess some natural image properties preserved within them even after the style transfer. When trained using such images, the network *SIN* shows improved shape bias as well as improved robustness against corruptions. This implies that a kind of style augmentation, which preserves some of the natural image properties, helps to boost the corruption robustness.

The network *E* is trained using edge maps that are devoid of any natural images statistics. Also, these edge maps are not stylized. *E* trained on such edge maps has high shape bias but shows poor performance on distortions. This indicates that the edge maps alone cannot

contribute to improved corruption robustness. Another network *SE* is trained using stylized edges that do not possess any natural image properties but had undergone stylization. This network exhibits high shape bias while at the same time has improved distortion accuracies when compared to *E* and *IN*. This shows the significance of stylization in improved corruption robustness. However, the distortion performance of *SE* is still lower than that of *SIN*. Such a difference indicates that the preserved natural image statistics played an important role in improved corruption robustness of *SIN*.

Finally, the network *SE+IN* trained using superposition images, which possess all the three considered image factors within them, has improved corruption robustness and it slightly outperforms *SIN* too. Despite having the highest mean Corruption Accuracy among all other networks, *SE+IN* doesn't possess high shape bias. It can be seen from Table 6.8 that *SE+IN* has the lowest shape bias among all other networks except *IN*. This shows that there is no clear correlation between the shape bias of a network and its robustness against common corruptions. Instead, the common factors among the two networks *SIN* and *SE+IN* are that they both possess properties of natural images but at the same time are strongly distorted with the help of stylization. Hence, such powerful data augmentation methods, which stay close enough to the data manifold but at the same time induce high diversity with appearance, seem to enforce the networks to learn more robust representations. Such robust representations need not necessarily indicate the shape, but they help the networks to gain improved corruption robustness.

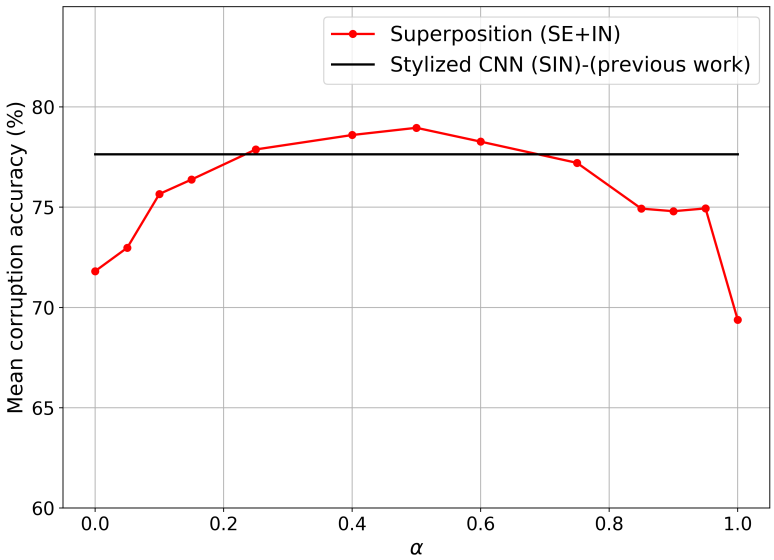

Figure 6.10: Mean corruption accuracy of superposition network variants (SE+IN) trained using images with different α values.

It is familiar that the superposition images are generated through the interpolation between

stylized edges and original images. As already discussed, the weightage between these two images is determined by an interpolation parameter α. Different values of α would result in different variants of superposition images. Hence, additional experiments are carried out to understand the significance of α in corruption robustness. Figure 6.10 contains the plots of the mean corruption accuracies of different *SE+IN* network variants trained using superposition images generated at different α values. The black line represents the mean distortion accuracy of the base network *SIN*. In this graph, $\alpha = 0$ corresponds to the stylized edges that are devoid of natural image properties. At the same time, α=1 represents natural images without any stylization. It can be seen from Figure 6.10 that for the α values ranging between 0.25 and 0.65, the corruption robustness of *SE+IN* is better than that of *SIN*. This shows that a proper weightage between stylization and preserved image statistics is necessary to ensure high robustness on ImageNet-C distortions. Further, even at an α value of 0.95, the mean corruption accuracy remains around 75%. However, for such a high value of α, the network exhibits lower shape bias with higher texture bias. This again shows that the shape bias has very little to do with the corruption robustness.

6.2.8 On the Adaptability of Learned Representations

It has been concluded in the previous subsection that effective style augmentation techniques can help the networks in adapting to data from different domains like ImageNet-C corruptions. In this section, a study is carried out to understand how easily the pretrained networks can be adapted to different data distributions. For this purpose, the data from different ImageNet-C corruptions are considered. Here, the pretrained networks are finetuned using the additional data from unknown distortions as well. Though such training cannot be used for the practical purpose of improving the corruption robustness, these experiments might help in understanding the internal representations of a network. A key factor to be considered in this experimental setting is that, during the finetuning on respective distortions, all other network parameters except the affine parameters of normalization layers are frozen. This means that the pretrained feature extractors are not adapted to the respective distortions. Instead, they were determined only using natural images.

The above-described approach is similar to the proposal of Chang et al. in [65], where the authors used domain-specific normalization parameters for adapting the networks to different domains. In their approach, all other network parameters except the affine parameters of normalization layers are shared over the different data distributions. However, the affine parameters are kept domain-specific. This means that the feature extractors are trained on the target distribution data too. Contrasting to this, in our experiments, the feature extractors are never exposed to the target distribution data while only the normalization parameters are updated on them. During our finetuning on ImageNet-C distortions, the severity levels are set to be randomly chosen by the network itself. On a standard pretrained network, the target distribution data are fed for 50 epochs with an initial learning rate of 0.01 which is then reduced to 0.001 at 45th epoch.

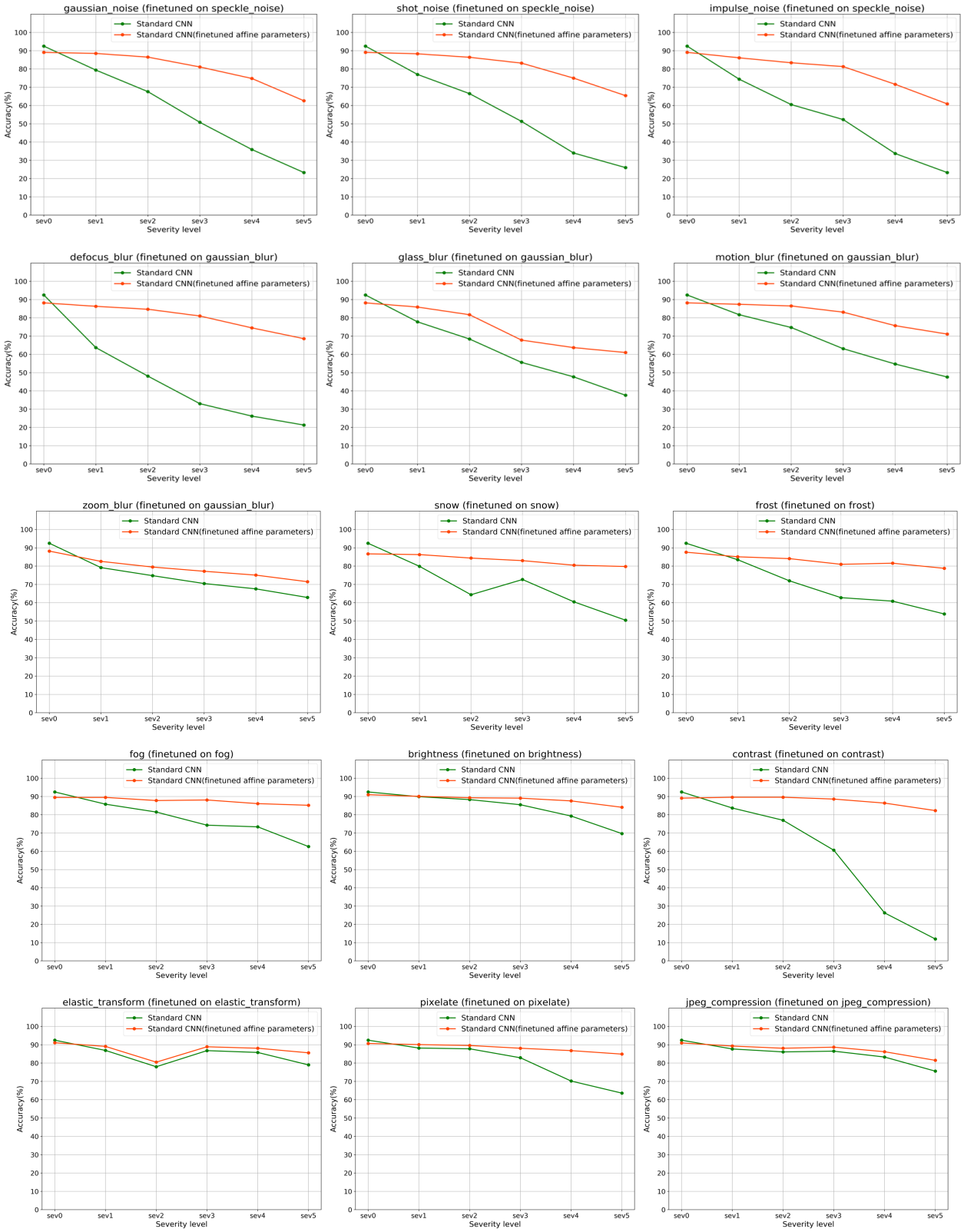

Figure 6.11: Performance of standard network on ImageNet-C corruptions along with its finetuned variants on respective distortions.

Figure 6.11 contains the plots for 15 different ImageNet-C distortions where the distortion accuracy of the standard network is plotted in green. The additional plot in red represents the distortion accuracies of those networks whose affine parameters are finetuned on respective distortions. For noise category, an improvement in distortion accuracies is observed for all different noises when the network is finetuned only on Speckle Noise. Similarly, for the blur category, the network is just finetuned on Gaussian Blur but it also shows improved performance on all other blur types. For other distortions, the networks are finetuned and evaluated on the respective distortions. The results from Figure 6.11 indicate that finetuning only the affine parameters of a network on the target data drastically improves its performance.

In addition to the standard network (*IN*), such finetuning is carried out for our shape based network (*E*) too. Also, other than ImageNet-C distortions, the affine parameter finetuning for these networks is carried out on Stylized-ImageNet (SIN) dataset as well. These results are presented in Table 6.9. This table represents the original and finetuned mean distortion accuracies of *IN* and *E* on the validation distortions of ImageNet-C. It could be observed that for any kind of distortions, both the networks show improved performance when their affine parameters are finetuned. These results thus indicate that the networks have already learned the internal representations that are robust to corruptions except that the affine layers did not leverage them.

	Corruptions (%)					**Cue Conflict**	
Network	**Speckle Noise**	**Gaussian Blur**	**Spatter**	**Saturate**	**SIN val acc(%)**	**shape #400**	**texture #100**
IN	61.28	42.96	72.94	74.3	42.0	63	39
IN (fine-tuned)	82.7	77.3	84.9	85.26	68.0	130	13
E	67.76	44.48	66.54	84.1	62.3	193	15
E (fine-tuned)	80.18	71.74	76.66	86.7	72.4	222	9

Table 6.9: Corruption and SIN accuracies along with the cue conflict results for the networks with and without finetuning of their affine parameters on respective datasets.

Further, the last three columns of the Table 6.9 represent the results on SIN dataset. Similar to distortions, both the networks show improved accuracy on the SIN dataset when their affine parameters alone are finetuned. It is further evident from the last two columns of Table 6.9 that, such a finetuning of normalization parameters using SIN data also shows significant improvement in the shape representations of the networks with a reduction in their texture bias. Particularly the results of *IN* and its finetuned version show that the standard network trained on ImageNet data already encodes significant shape details that get leveraged when their affine parameters are finetuned using complex stylized data such as SIN.

7 Conclusion and outlook

This chapter provides a summary of the thesis work. Further, it also discusses a short note on the limitations of the proposed work and the possibilities for future research.

7.1 Conclusion

7.1.1 Enhancing the Shape Bias of CNNs

Recent studies revealed that the CNNs, which have state-of-the-art results in many image manipulation tasks, exhibit texture bias. On the other hand, humans are biased towards the object shape. In order to bridge this behavioral gap between humans and CNNs, improving the shape bias of CNNs is imperative. Motivated by this, an approach that utilizes edge maps to enhance the shape bias of CNNs is proposed in this thesis work. The edge maps, which represent the contours of the images, carry explicit shape details of the objects. Such edge maps could therefore help in the shape based decision making of CNNs.

An existing edge detection network named RCF [11] is utilized to extract the edge details from the standard images. Different variants of edge maps are constructed and their results are analyzed to select the prominent one for implementing the shape based network. A dataset called ImageNet20, which contains a set of 20 handpicked classes from the ImageNet1000 dataset is utilized for conducting the experiments. Further, the network architecture considered is ResNet18. The training of the shape based network is carried out in two stages. An initial training that uses only the edge maps to train the network. This is followed by the finetuning stage, where the edge maps and the original RGB images are together fed to the network. A shape based network trained using the above setting shows improved shape bias when compared to a standard CNN trained only on natural RGB images. This implies that the edge maps forced the network to focus on the object shape rather than other local details.

Further, as our shape based network is exposed to the standard RGB images during finetuning, it may have learned to encode their texture details. To reduce this texture dependency, a method called *style randomization* is introduced in this thesis work. Style randomization randomizes the style information in feature space by modifying the statistics of feature maps. The values for the feature statistics are chosen randomly from a uniform distribution. A CNN that implements style randomization is shown to have enhanced shape bias when compared to a network without any stylization.

7.1.2 Evaluation of Shape Bias

The shape bias of a network is evaluated using different evaluation strategies namely, patch shuffled images, and texture-shape cue conflict images. Patch shuffled images are formulated by splitting an image into different patches and then randomly shuffling those patches before rejoining. By doing so, the texture details encoded in the images get preserved while the global object structure is destroyed. Hence, a network with more shape bias may fail to recognize such images. The texture-shape cue conflict images are generated by transferring the style of one object to the structure of another object. Therefore, every such image possesses two labels namely, the texture label and the shape label. A network with high shape bias is expected to classify these cue conflict images according to their shape labels. On the other hand, a texture-biased CNN may predict them based on their texture labels. On evaluating the shape based network proposed in this thesis work with the above two datasets, it is shown to outperform the existing baselines.

7.1.3 Shape Bias doesn't Improve Corruption Robustness

Geirhos et al. [1] introduced a dataset termed Stylized ImageNet (SIN), which is obtained through the stylization of natural images, to improve the shape bias of CNNs. A network trained on such stylized images is shown to exhibit stronger shape bias. Further, when such a network is evaluated on ImageNet-C corruptions it shows improved robustness. The enhanced shape bias of the stylized network is assumed to be the reason behind its improved corruption robustness. However, the shape based network proposed in this thesis work doesn't show any performance improvement on ImageNet-C corruptions though it has stronger shape bias than the stylized network. This motivated us to systematically analyze if there exists any clear correlation between the shape bias of a network and its robustness against corruptions. For this purpose, various networks that are trained on different stylization variants are evaluated on ImageNet-C distortions. The corresponding results revealed that the improved robustness of the CNN trained on stylized images comes from the following fact. Stylization refers to a strong form of data augmentation which forces the network to learn robust representations regardless of whether such representations are shape cues. These robust representations help in improving the corruption robustness of the network. The enhanced shape bias, which is attained through such an augmentation, could therefore be considered as an unrelated byproduct. This indicates that enhancing the shape bias of a network need not improve its corruption robustness and hence, shape based learning is not a direction to explore when the primary focus is to improve the robustness against corruptions.

7.1.4 Adaptability of Learned Representations

In addition to the above experiments, a study is conducted to understand how easily a pretrained network can adapt to different data distributions like corruptions. To do so, a network that is pretrained on standard images is finetuned on the target data for considerable iterations. However, this finetuning is done only for the affine parameters of the normalization

layers. It is shown that such a finetuned network shows good performance improvement on the target dataset. Such results imply that the ImageNet trained CNNs have already learned the feature representations that are robust to different data distributions. However, these representations are not leveraged in their affine layers. When these affine parameters of normalization layers are tuned properly, a standard texture-biased network shows improved shape bias.

The proposed approach for shape based learning shows higher shape bias than existing approaches. By detailed analysis, it is also shown in this work that the shape bias of a network need not improve its corruption robustness. However, an application for this shape biased network is not yet explored. Such possible future extensions of this work are discussed in the following subsection.

7.2 Outlook

Our experiments have demonstrated that the enhanced shape bias of a CNN need not necessarily improve its corruption robustness. However, the shape learning bridges one of the existing gaps between human visual perception and the object perception by CNNs. Hence, although not useful for corruption robustness, there could be potential applications for this shape based learning. Future research directions can focus on exploring such possible applications of shape based learning. For example, a recent work by Zhang et al. [9] showed that adversarial training improves the shape bias of a network. Therefore, it would be interesting to analyze if the vice-versa, i.e. improving the shape bias of a network improves its adversarial robustness, holds true. Further, considering the results of various stylization variants on corruption robustness, another possible direction would be to devise stronger data augmentation techniques either in image space or in feature space for improving the distortion robustness. Finally, a clear understanding of the features that play an essential role in improving the corruption robustness of a CNN is very much required and hence could be an interesting new direction of research.

Bibliography

[1] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint arXiv:1811.12231*, 2018.

[2] N. Baker, H. Lu, G. Erlikhman, and P. J. Kellman, "Deep convolutional networks do not classify based on global object shape," *PLoS computational biology*, vol. 14, no. 12, p. e1006613, 2018.

[3] Y. Wu and K. He, "Group normalization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 3–19, 2018.

[4] X. Huang and S. Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1501–1510, 2017.

[5] S. Qiao, H. Wang, C. Liu, W. Shen, and A. Yuille, "Weight standardization," *arXiv preprint arXiv:1903.10520*, 2019.

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

[7] T. Luo, T. Cai, X. Zhang, S. Chen, D. He, and L. Wang, "Defective convolutional layers learn robust {cnn}s," 2020.

[8] H. Nam, H. Lee, J. Park, W. Yoon, and D. Yoo, "Reducing domain gap via style-agnostic networks," *arXiv preprint arXiv:1910.11645*, 2019.

[9] T. Zhang and Z. Zhu, "Interpreting adversarially trained convolutional neural networks," *arXiv preprint arXiv:1905.09797*, 2019.

[10] 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.

[11] Y. Liu, M.-M. Cheng, X. Hu, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3000–3009, 2017.

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

[13] B. Landau, L. B. Smith, and S. S. Jones, "The importance of shape in early lexical learning," *Cognitive development*, vol. 3, no. 3, pp. 299–321, 1988.

[14] R. Geirhos, C. R. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann, "Generalisation in humans and deep neural networks," in *Advances in Neural Information Processing Systems*, pp. 7538–7550, 2018.

[15] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[16] S. Santurkar, D. Tsipras, A. Ilyas, and A. Madry, "How does batch normalization help optimization?," in *Advances in Neural Information Processing Systems*, pp. 2483–2493, 2018.

[17] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[18] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[19] J. Koenderink, M. Valsecchi, A. van Doorn, J. Wagemans, and K. Gegenfurtner, "Eidolons: Novel stimuli for vision research," *Journal of Vision*, vol. 17, no. 2, pp. 7–7, 2017.

[20] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[21] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2574–2582, 2016.

[22] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," *arXiv preprint arXiv:1611.01236*, 2016.

[23] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

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

[25] 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*, pp. 770–778, 2016.

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1–9, 2015.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[28] J. Canny, "A computational approach to edge detection," *IEEE Transactions on pattern analysis and machine intelligence*, no. 6, pp. 679–698, 1986.

[29] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2414–2423, 2016.

[30] W. Brendel and M. Bethge, "Approximating cnns with bag-of-local-features models works surprisingly well on imagenet," *arXiv preprint arXiv:1904.00760*, 2019.

[31] J. Jo and Y. Bengio, "Measuring the tendency of cnns to learn surface statistical regularities," *arXiv preprint arXiv:1711.11561*, 2017.

[32] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[33] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on knowledge and data engineering*, vol. 22, no. 10, pp. 1345–1359, 2009.

[34] S. Ben-David, T. Lu, T. Luu, and D. Pál, "Impossibility theorems for domain adaptation," in *International Conference on Artificial Intelligence and Statistics*, pp. 129–136, 2010.

[35] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1406–1415, 2019.

[36] H. Wang, S. Ge, Z. Lipton, and E. P. Xing, "Learning robust global representations by penalizing local predictive power," in *Advances in Neural Information Processing Systems*, pp. 10506–10518, 2019.

[37] K. L. Hermann and S. Kornblith, "Exploring the origins and prevalence of texture bias in convolutional neural networks," *arXiv preprint arXiv:1911.09071*, 2019.

[38] F. B. TREES, "Forest before trees: The precedence of global features in visual perception," *Cognitive psychology*, vol. 353, p. 383, 1977.

[39] H. Hosseini, S. Kannan, and R. Poovendran, "Dropping pixels for adversarial robustness," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[40] R. Huang, B. Xu, D. Schuurmans, and C. Szepesvári, "Learning with a strong adversary," *arXiv preprint arXiv:1511.03034*, 2015.

[41] A. Nøkland, "Improving back-propagation by adding an adversarial gradient," *arXiv preprint arXiv:1510.04189*, 2015.

[42] Y. Le and X. Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, 2015.

[43] G. Griffin, A. Holub, and P. Perona, "Caltech-256 object category dataset," *CalTech Report*, 2007.

[44] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, vol. 55, 2014.

[45] R. G. Lopes, D. Yin, B. Poole, J. Gilmer, and E. D. Cubuk, "Improving robustness without sacrificing accuracy with patch gaussian augmentation," *arXiv preprint arXiv:1906.02611*, 2019.

[46] 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*, pp. 6023–6032, 2019.

[47] E. D. Cubuk, B. Zoph, J. Shlens, and Q. V. Le, "Randaugment: Practical automated data augmentation with a reduced search space," *arXiv preprint arXiv:1909.13719*, 2019.

[48] Q. Xie, E. Hovy, M.-T. Luong, and Q. V. Le, "Self-training with noisy student improves imagenet classification," *arXiv preprint arXiv:1911.04252*, 2019.

[49] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7132–7141, 2018.

[50] R. Zhang, "Making convolutional networks shift-invariant again," *arXiv preprint arXiv:1904.11486*, 2019.

[51] D. Hendrycks, M. Mazeika, S. Kadavath, and D. Song, "Using self-supervised learning can improve model robustness and uncertainty," in *Advances in Neural Information Processing Systems*, pp. 15637–15648, 2019.

[52] 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.

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

[54] Y. Grandvalet and S. Canu, "Noise injection for inputs relevance determination," in *Advances in intelligent systems*, pp. 378–382, IOS Press, 1997.

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

[56] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation strategies from data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 113–123, 2019.

[57] D. R. Martin, C. C. Fowlkes, and J. Malik, "Learning to detect natural image boundaries using local brightness, color, and texture cues," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, no. 5, pp. 530–549, 2004.

[58] P. Dollár and C. L. Zitnick, "Fast edge detection using structured forests," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 8, pp. 1558–1570, 2014.

[59] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.

[60] G. Bertasius, J. Shi, and L. Torresani, "Deepedge: A multi-scale bifurcated deep network for top-down contour detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4380–4389, 2015.

[61] W. Shen, X. Wang, Y. Wang, X. Bai, and Z. Zhang, "Deepcontour: A deep convolutional feature learned by positive-sharing loss for contour detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3982–3991, 2015.

[62] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, pp. 1395–1403, 2015.

[63] V. Dumoulin, J. Shlens, and M. Kudlur, "A learned representation for artistic style," *arXiv preprint arXiv:1610.07629*, 2016.

[64] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT'2010*, pp. 177–186, Springer, 2010.

[65] W.-G. Chang, T. You, S. Seo, S. Kwak, and B. Han, "Domain-specific batch normalization for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7354–7362, 2019.

TECHNISCHE UNIVERSITÄT CHEMNITZ

Studentenservice – Zentrales Prüfungsamt

Selbstständigkeitserklärung

Name: Subramaniam Vorname: Ranjitha geb. am: 08.10.1991 Matr.-Nr.: 516518	Bitte beachten: 1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende Masterarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: 23.06.2020 Unterschrift: .....

---

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.