SEMINAR REPORT

On the Importance of Shape and Texture Bias for Object Recognition

Stefan-Daniel Vilceanu

stefan.vilceanu@rwth-aachen.de

Matrikelnummer: 407531

Supervisor: Azad Reza

June 3, 2023

Contents

1 Introduction

Artificial intelligence has vastly grown in the last decades thanks to the evolution of technology. Artificial intelligence focuses on creating intelligent machines capable of performing human tasks. One of its main and most researched areas is represented by machine learning. Machine Learning consists of algorithms and mathematical models that give the ability to a computer to interpret, analyze, make predictions, and decisions based on the provided experience gained through complex data. Probably the most interesting and prominent models of machine learning are represented by artificial neural networks (ANN). These are computational models inspired by the biological neural networks in the human brain. ANNs consist of nodes, organized in layers, and weights, receive inputs and produce outputs, but the crucial part is that these networks have the ability to learn by training them with given data and by adjusting their weights.

The topic of this report relies on a specific type of ANN, namely Convolutional Neural Network (CNN). This type of network is particularly good at processing and analyzing structured grid-like data, such as images and videos. The main tasks of a CNN are object detection, image classification or object recognition. In contrast to traditional neural networks, CNNs use convolutional layers that learn and extract patterns from input data. The functionality of these layers consists of kernels that convolve over the input, e.g., pixels of an image, in order to obtain key features. Because nowadays almost every image has a substantial resolution, pooling layers are used in CNNs to down-sample the dimensions of a given feature map in order to significantly reduce computation tasks. At later stages in the network, fully connected layers are used as well as an activation function for classification purposes.

Another focus of this report is held upon an alternative solution to CNNs, called Vision Transformer (ViT). A ViT is also a type of neural network architecture that uses the transformer model in the field of computer vision [1]. The working mechanism of a ViT consists of dividing an input image into non-overlapping patches, embedding these, and inserting them into the transformer encoder. By using a self-attention mechanism, attention maps are calculated. These give the relationship and contextual information between each patch and all the other patches of the input image. Afterwards, the classification output can be calculated using a feed-forward neural network (e.g multi-layer perceptron). The architecture of such a Vision Transformer can be seen in Figure 1. The lower part of this figure presents the division of the input image into patches which are then input into the transformer encoder. The right-hand side of the figure shows the architecture inside it.

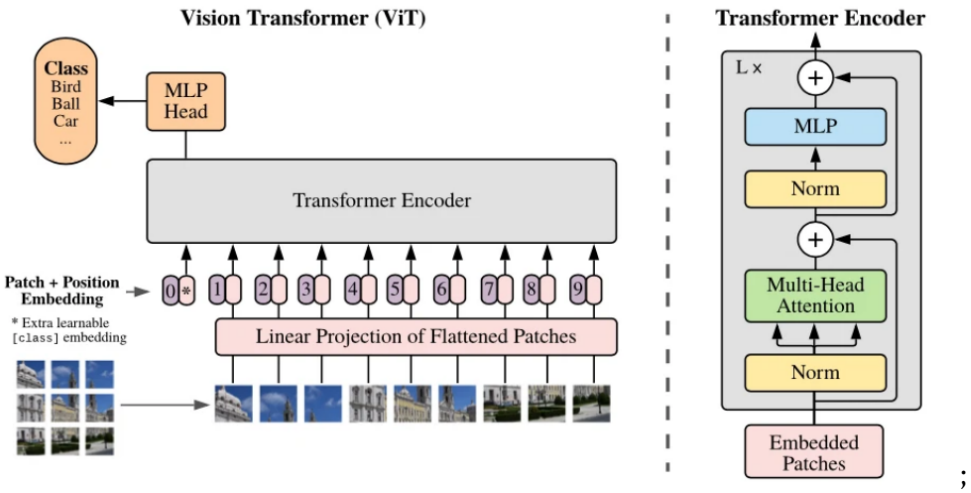


Figure 1: Vision Transformer Architecture

This report aims to create an overview of some of the encountered problems when using Convolutional Neural Networks and Vision Transformers but also to present several solutions that were engineered to solve these issues.

Generally, it was taught that CNNs identify objects and learn their features by studying the complex representations of object shapes, but according to [2], CNNs actually behave somewhat differently. In this paper, it is demonstrated that CNNs trained on ImageNet are actually significantly more biased upon learning the textures of images rather than acquiring knowledge of the shape of the objects [3]. Experiments were produced by comparing the classification results of CNNs with the acquired results of human observers. The comparison was made by analyzing images with texture-shape cue conflicts [4]. This means that new images were generated by combining normal images from a data set with a new style that would change the texture of the original image. Figure 2 presents such an example, where the image of a cat is mixed with the style of another picture that shows the skin of an elephant. The result presents a cat image with a different texture. Under the three images, the decision confidence of the CNN is shown, the first two images are being correctly classified, while the third one is being miss classified due to the texture bias that CNNs exert.

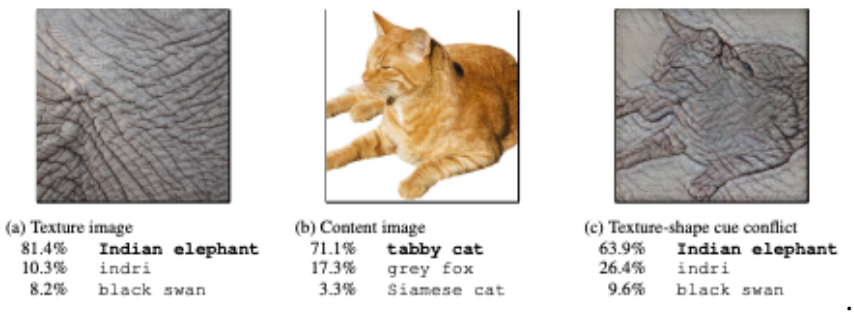
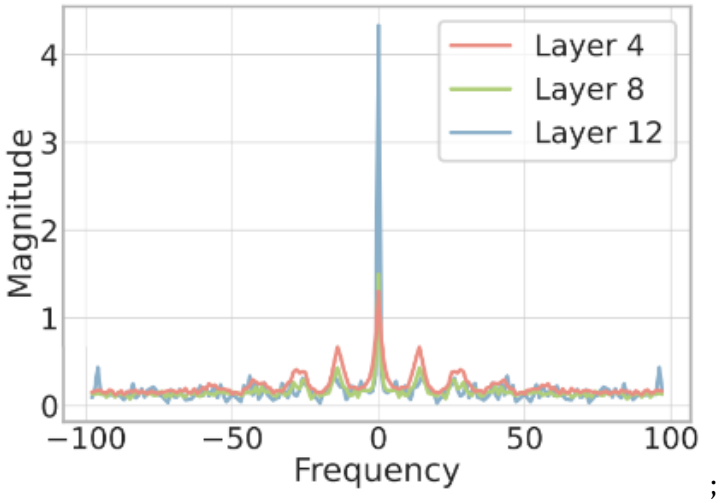


Figure 2: Cue conflict cat image

The second presented issue in this report tackles the over-smoothing effect present in Deep Vision Transformers. Paper [5] discusses and proves that the self-attention mechanism is, in fact, a low-pass filter. A low-pass filter deletes high-frequency details from the input patches which creates the over-smoothing effect and reduces the features' expressiveness at deep layers. Deleting high-frequency information actually means removing information about the edges and color intensity, which increases shape and decreases texture bias. Fortunately, two techniques are presented for this problem. The first method, called Attention Scaling (AttnScale), translates the self-attention block into low-pass as well as high-pass filters and manipulates them in order to produce an all-pass self-attention mechanism. The second technique, named Feature Scaling (FeatScale), re-weights feature maps on separate frequency bands to enhance the high-frequency signals.



;

Figure 3: Spectral response of an attention map

Figure 3 shows the frequency spectrum of an attention map at three different layers. It can be observed that the deeper the ViT goes, the more discrepancies there are between the direct current component (DC) and the high-frequency component (HC) of the signal. While the ViT still preserves some of the high-frequency details at layer four, when it comes to the 12-th layer, the attention map becomes almost an ideal low-pass filter, smoothing out a great amount of HC information.

2 Methods

This chapter will present in more detail the three proposed methods described in the introduction section.

2.1 Stylized-ImageNet

The method proposed by [2] is based on the comparison of multiple experiments between the performance of humans and CNNs on correct classifications. The four CNNs, namely AlexNet, GoogLeNet, VGG-16, and ResNet-50, were trained on the standard ImageNet library. The experiment consists of images of 16 different classes: airplane, bear, bicycle, bird, boat, bottle, car, cat, chair, clock, dog, elephant, keyboard, knife, oven, and truck, and comes with four different features added to the original image, these being grey-scale, silhouette, edges, texture and cue conflict. These initial tests are experimented with in order to prove the hypothesis that CNNs do, in fact, have texture biases present and in order to asses them. Figure 4 shows five images with five different features and demonstrates the performance falloff of the CNNs when textures are being removed. The last image reiterates the point of the hypothesis where it can be seen that a texture-based image can easily be correctly classified by all the CNNs. The grey-scale feature was created by mapping the color intensities to be white, black, and in between. The silhouette shows the whole object of interest in black, while the background becomes white. The edge representations were realized using a MATLAB package, and finally, the texture images were gathered from the skin or fur of animals or were created by putting multiple man-made objects very close to each other.

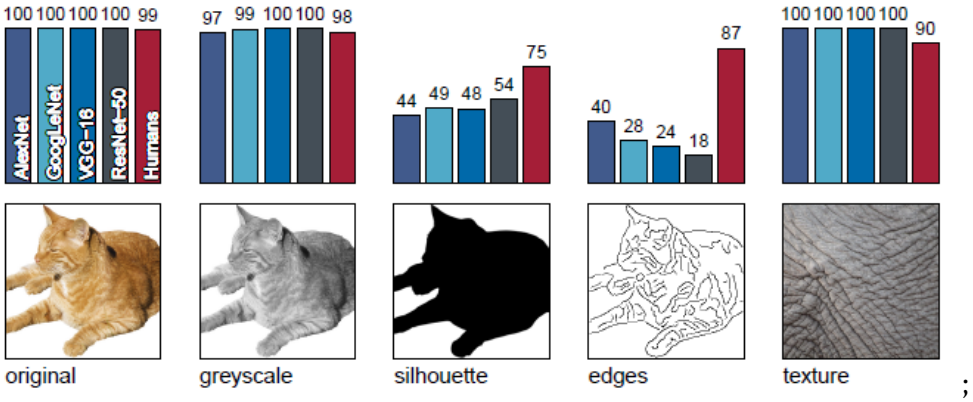


Figure 4: Performance comparison of images with different features

To solve this problem, [2] creates a new data set and uses the original images of the ImageNet library by changing their style. This new data set is called Stylized-ImageNet (SIN),and it was created using the AdaIN Neural Style Transfer package of Keras. The styles used were taken from selected artistic paintings of the Kaggle's Painter by Numbers data set, and the new resulting images realize the idea of cue conflict examples. The purpose of this data collection

is to ensure less biased training for the CNN. Figure 5 exemplifies such a style transfer. The left image is an example from the ImageNet data set, and next to it, there are ten different interpretations of it, all using themes from random paintings. This approach ensures the removal of local texture features and increases shape bias, which obligates the CNNs to look further into the image for correct image classification, instead of rapidly interpreting the texture details [6].

;

Figure 5: Example of style transfer

2.2 AttnScale

As mentioned in the introductory chapter, the ViT struggles with the over-smoothing effect caused by the self-attention mechanism. In contrast to CNNs, the ViT presents a shape bias seen from a frequency point of view by capturing long-range dependencies of images and partially missing some of their local features. Thus, it can be presumed that the ViT only keeps the DC component of its frequency spectrum. [5] introduced the so-called Attention Scaling technique as a mean to undermine the smoothing effect presented in ViTs. The functioning principle of the AttnScale is to separate the self-attention matrix into a low-pass filter and a high-pass filter, which ideally should result in having an all-pass filter plus to introduce a scaling parameter for the high-pass filter for magnitude matching. Using this reworked all-pass filter, a new self-attention matrix can be formed. Applying this approach would result in a lower removal rate of high-frequency information at higher layers of the transformer. The left side of Figure 6 shows the implementation of the AttnScale technique as a block diagram. The query, $\boldsymbol{Q}$, and key, $\boldsymbol{K}$, vectors are first multiplied, then their output is applied on a SoftMax function, finally receiving a low-pass and a high-pass self-attention matrix filter, with ω being the trainable scaling parameter.

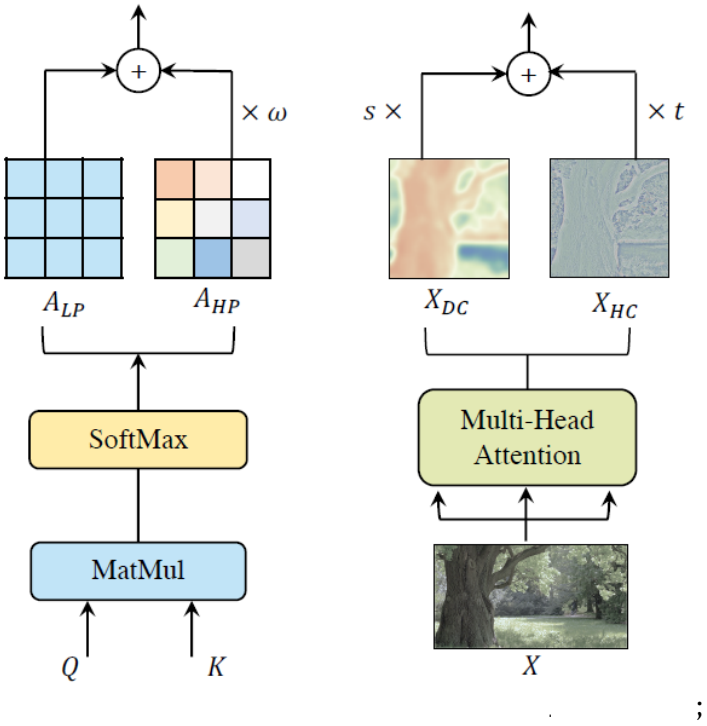


Figure 6: Block diagram of Attn- and FeatScale

2.3 Featscale

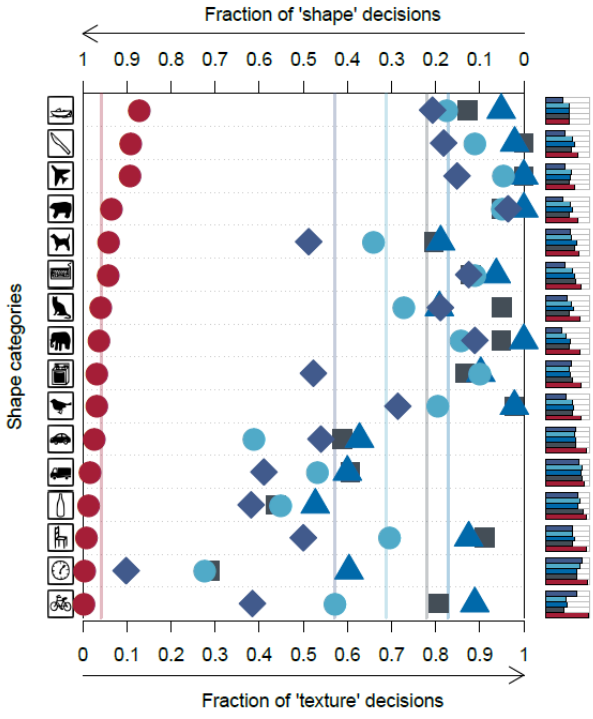
The second anti-smoothing technique proposed by [5] is the FeatScale method. In a brief description, the FeatScale integrates a re-weighting mechanism of the feature maps on different frequency intervals in order to boost the high-frequency signals. Because the multi-head self-attention block (MSA) suppresses the high-frequency signals, the FeatScale is designed to manipulate the MSA output by modifying the information from multiple frequency ranges distinctively. The manipulation happens by separating the resulting signals into the two frequency components and then, by adding two groups of trainable parameters, it can re-weight the DC and HC components of the signal for each frequency range. The right diagram block from Figure 6 shows the described implementation. The feature map $\boldsymbol{X}$ goes through the MSA, which separates its output in DC and HC feature maps, which then can be re-weighted with the parameters $\boldsymbol{s}$ and $\boldsymbol{t}$.

3 Results

The following section will present an overview of the results gathered using the previously mentioned methods.

3.1 Performance comparison before and after texture bias

Looking again at Figure 4, the texture bias in CNNs can be easily observed when looking at the classification performance when texture features are removed, such as in the case of silhouette and edge images. The comparison between human and CNN performance was also looked at from a different perspective. Figure 7 shows the deciding factor, texture or shape, when an image has to be classified. The red dots represent the fraction of the human observers, which is significantly biased towards shape classification, while blue palette-colored circles (GoogLeNet), diamonds (AlexNet), squares (ResNet-50), and triangles (VGG-16) represent the results of the CNNs. Here, the texture bias of the CNN can be observed in some categories, this being almost 100% of the deciding factor for classifications. The results from this figure were gathered by testing on cue conflict stimuli of the ImageNet data set. Percentage-wise, the classification results were as follows. The least biased CNN was AlexNet, with 42.9% shape decision vs. 57.1% texture decision, followed by GoogLeNet with 31.2% vs. 68.8%, ResNet-50 with 22.1% vs. 77.9% and VGG-16 with 17.2% vs. 82.8%.



;

Figure 7: Classification results between human observers and four CNNs

As presented in the method section, the advertised solution was to create a new ImageNet

data set, where the style of the original image needed to be changed with randomly selected paintings. A ResNet-50 CNN was therefore trained on this newly created Stylized-ImageNet data set. Table 1 shows the comparison in classification performance between two ResNet-50 CNNs, one trained on IN, the other on SIN when it comes to testing itself on a data set. For example, IN → SIN means training on ImageNet and testing on Stylized-ImageNet. On the one hand, this data shows that training and testing on IN are much easier than on SIN, as the IN shows a 92.9% accuracy, while the SIN is only 79.0% accurate. On the other hand, it can be observed that ResNet-50 struggles when it is tested on SIN after being trained on IN, achieving a very low accuracy, while the other way around, the SIN-trained CNN demonstrates great results upon being evaluated on the IN data set.

Architecture	IN → IN	IN → SIN	SIN → SIN	SIN → IN
ResNet-50	92.9%	16.4%	79.0%	82.6%

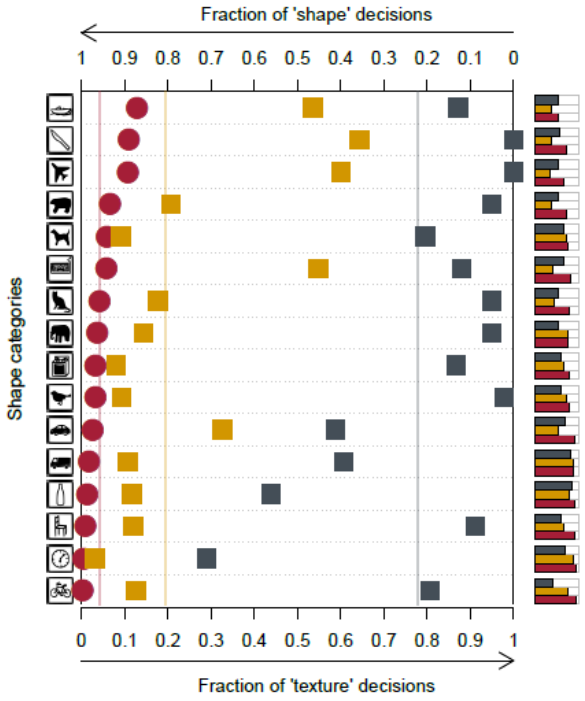
Table 1: Testing performance between two ResNet-50, one trained on IN, the other on SIN

The same concept from Figure 7 was applied once again to Figure 8. Here only two ResNet-50 CNNs were compared to the human observers. The grey colored squares represent the IN-trained CNN, and the yellow ones the SIN-trained network when tested on cue conflict images. The red circles are once again the data from the observers. The main idea presented here is that the SIN-trained CNN managed to eliminate its texture bias, even more, in some categories, it had almost the same shape deciding factor as the human observers. Overall, the ResNet-50, when trained on the SIN data set, increased its shape bias from 22.1% all the way up to 81%.

3.2 Performance results regarding anti-oversmoothing

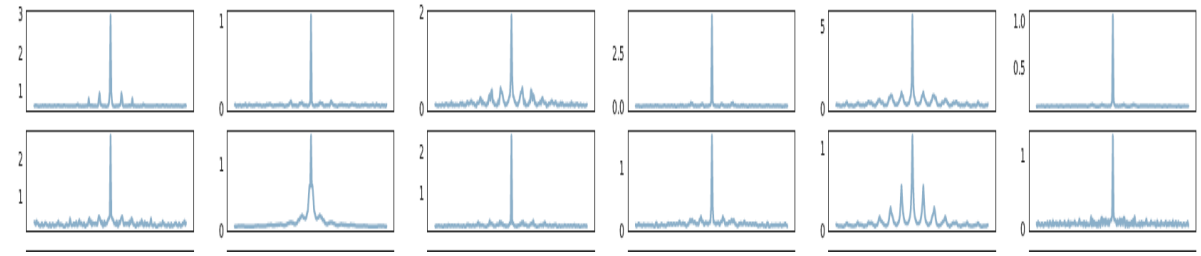
Multiple experiments were conducted using the two anti-oversmoothing techniques, AttnScale and FeatScale, exaplained in the previous chapter. Three backbones are used, namely DeiT, CaiT and Swin-Transformer and both these methods were applied to the patch embedding layers [7][8]. As a performance overview, the FeatScale slightly edges AttnScale out. Looking at the experiments with 24 layers, in the case of DeiT, the top-1 accuracy was 80.5%. Using DeiT with AttnScale increased its accuracy up to 81.1%, while applying FeatScale it achieved 81.3%. Regarding CaiT, the top-1 accuracy was 82.6% and when utilizing FeatScale or AttnScale, the accuracy raised to 83.2% for both cases. The Swin-Transformer showed an 83.0% accuracy and when combined with AttnScale and FeatScale, it achieved 83.4%, respectively 83.5% top-1 accuracy.

Figure 9 shows a visualization of the effect when using AttnScale. Looking at the first image, the local texture information exposed through high-frequency signals is clearly faded, having almost only the DC left. In contrast, after using AttnScale, the second image shows the

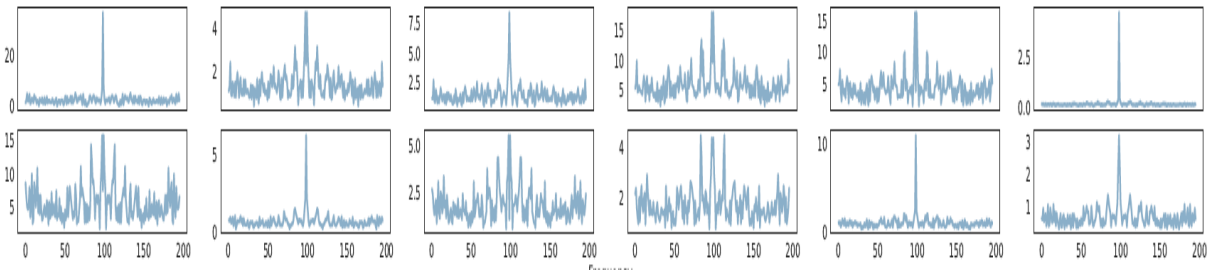


;

Figure 8: Classification results between human observers and two ResNet-50, one trained on SIN, the other on IN



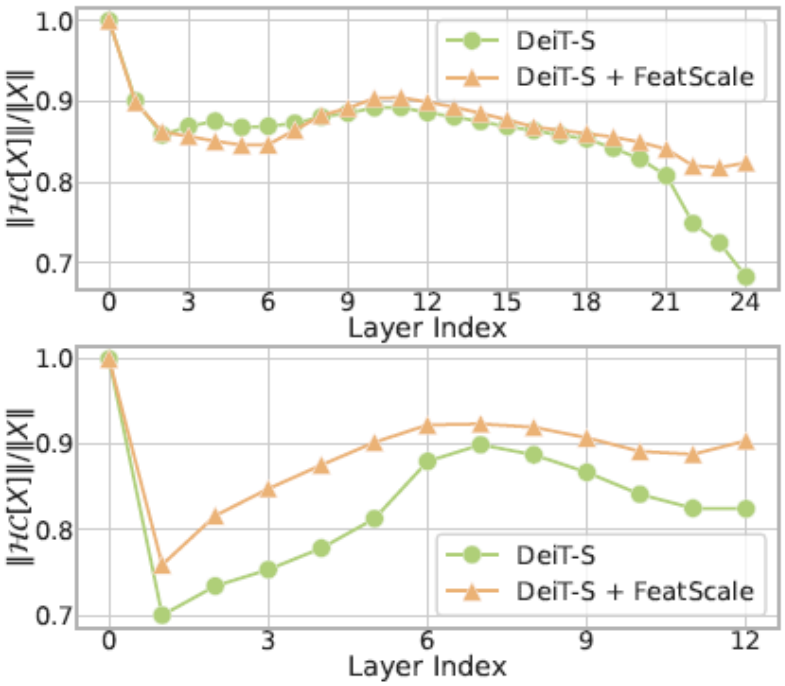
(a) without AttnScale



(b) with AttnScale

Figure 9: Spectrum of attention maps at the deepest two layers. y-axis represents magnitude; x-axis frequency

up-scaling of the HC, so that even at the deepest layers of the transformer the local texture information is not lost.



;

Figure 10: Visualization of the high-frequency proportion during the layers

Figure 10 shows the comparison of the high-frequency component preserved during the layers of a ViT with and without FeatScale. When utilizing a 12-layer ViT with FeatScale, the HC proportion is consistently greater than when utilizing a simple DeiT-S, but the biggest difference is presented when working with a much deeper transformer. In the case of a 24-layer ViT shown in the upper image of this figure, when FeatScale is not used, a great amount of HC is being dropped when being in the deepest layers. Thus, using this technique, the high-frequency signal can be substantially preserved.

4 Conclusion

This report presented an overview of the importance of shape and texture bias in the field of computer vision by reiterating some of the key ideas and the methodologies used from the given references. It consisted of an introductory chapter, where the motivation and general concepts were briefly described. The second section presented three methods for overcoming texture and shape biases when working with convolutional neural networks or vision transformers. Finally, some of the gathered results and comparisons were presented in the last chapter and the overall performance of the iterated methods has been assessed.

References

[1] A. Dosovitskiy, L. Beyer, A. Kolesnikov, *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

[2] 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.

[3] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture and art with deep neural networks," *Current opinion in neurobiology*, vol. 46, pp. 178–186, 2017.

[4] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[5] P. Wang, W. Zheng, T. Chen, and Z. Wang, "Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice," *arXiv preprint arXiv:2203.05962*, 2022.

[6] 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*, 2016, pp. 2414–2423.

[7] H. Touvron, M. Cord, A. Sablayrolles, G. Synnaeve, and H. Jégou, "Going deeper with image transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 32–42.

[8] C. Gong, D. Wang, M. Li, V. Chandra, and Q. Liu, "Vision transformers with patch diversification," *arXiv preprint arXiv:2104.12753*, 2021.