Shape Prior is Not All You Need: Discovering Balance between Texture and Shape bias in CNN

Hyunhee Chung*[1][0000−0002−0113−3126],
and Kyung Ho Park*[1][0000−0002−3439−9297]

SOCAR AI Research, Korea
{esther,kp}@socar.kr

Abstract. As Convolutional Neural Network (CNN) trained under ImageNet is known to be biased in image texture rather than object shapes, recent works proposed that elevating shape awareness of the CNNs makes them similar to human visual recognition. However, beyond the ImageNet-trained CNN, how can we make CNNs similar to human vision in the wild? In this paper, we present a series of analyses to answer this question. First, we propose AdaBA, a novel method of quantitatively illustrating CNN's shape and texture bias by resolving several limits of the prior method. With the proposed AdaBA, we focused on fine-tuned CNN's bias landscape which previous studies have not dealt with. We discover that fine-tuned CNNs are also biased to texture, but their bias strengths differ along with the downstream dataset; thus, we presume a data distribution is a root cause of texture bias exists. To tackle this root cause, we propose a granular labeling scheme, a simple but effective solution that redesigns the label space to pursue a balance between texture and shape biases. We empirically examine that the proposed scheme escalates CNN's classification and OOD detection performance. We expect key findings and proposed methods in the study to elevate understanding of the CNN and yield an effective solution to mitigate this texture bias.

1 Introduction

Discovering what Convolutional Neural Network (CNN) learned has become an important but challenging problem in modern computer vision studies[16,8,5,1]. Recent studies presented that CNNs, especially those trained under the ImageNet [3] dataset have texture bias that prioritizes image textures rather than object's shapes. This finding conflicts with human visual perceptions as humans utilize shape information to understand images [7]. The aforementioned texture bias of CNN is known to be a critical challenge due to the following reasons. First, CNN's texture bias might become a vulnerability from a security perspective as an adversary would attack the model by transforming the image's textures to mislead its understanding [14]. Second, CNN's texture bias might

* These authors contributed equally to this work.

exhibit its inductive bias being distinct from the human visual system, which indicates insufficient robustness to be deployed in the real world. Under the insufficient robustness of CNN, it risks creating fatal damage to the humans who interact with the model (i.e., medical imaging [20,22]).

While prior analyses presented monumental findings and solutions regarding CNN's texture bias, we figure out several improvement avenues. First, prior studies primarily focused on analyzing the CNN's dynamics in ImageNet and its derived ones (i.e., Cue-conflict dataset [7]). While fine-tuning has become a de facto technique in modern computer vision tasks (i.e., image recognition [18,24,11], object detection [2,21]), there were no bias analyses on the fine-tuned CNNs. To this end, we postulate several questions regarding the fine-tuned CNN'S dynamics: Do CNNs fine-tuned on various downstream datasets show texture bias, just as the case in ImageNet? If so, do fine-tuned CNNs show similar bias strength regardless of the downstream dataset?

Second, we urge that the frozen label space assumption should be released for a more practical solution against CNN's texture bias. We denote a frozen label space as the assumption that does not change the labeling scheme but uses the given dataset as it was originally labeled. As Hermann et al. [14] once proposed, a data distribution (as well as label space) of conventional datasets (i.e., ImageNet) is a root cause of CNN's texture bias [14]. Nevertheless, we analyze that previously-proposed solutions have not tackled this root cause but primarily focused on additional actions given a trained model [7,25]. Suppose the practitioners establish a labeled dataset from unlabeled samples before model training. What if we can mitigate the texture bias when the practitioners design a labeling scheme? What if we can resolve texture bias before model training? If the practitioners can resolve the texture bias by simply changing the labeling scheme, we expect it to become a powerful solution in the real world.

To accomplish these improvement avenues, our study proposes a series of analyses that scrutinize answers to the aforementioned questions. The contributions of our study are as follows. First, we seek a quantitative tool to analyze CNN's bias for an accurate understanding of its dynamics. While [15] once presented a solid baseline of this quantitative tool, we discovered several limits. Therefore, as an advanced version of the baseline, we propose a novel bias analysis method denoted Adaptive Bias Analysis (AdaBA), and empirically examine that our method coherently exhibits the same result as the previously-proposed method while it improves its limits. Second, we further analyze the dynamics of fine-tuned CNNs, which prior studies have not actively scrutinized. We analyze that fine-tuned CNNs are also biased to textures, but the strength of texture bias differs in downstream datasets. Thus, we presume a root cause of CNN's texture preference is indeed a data distribution, just as proposed in a recent discovery [14]. Third, we propose a novel viewpoint (problem setup) to mitigate CNN's texture bias by redesigning the labeling scheme before model training. We propose a Granular labeling scheme, a novel label space design to acquire a balance between texture and shape bias. Upon the synthetically-created datasets, we experimentally examine a CNN trained under the proposed granular labeling

scheme (which embraces the balance between texture and shape bias) is advantageous in two tasks: classification and out-of-distribution (OOD) detection. Lastly, we further analyze how the representation acquired under the granular labeling scheme differs from the others through measuring representation similarity with Centered Kernel Alignment (CKA) [17,10].

2 Related Works

As the original motivation of CNN's design stems from neuroscience [9,27], it has been regarded to recognize the image based on shape information, just as human perception[23,4]. However, Geirhos et al. empirically validates that CNNs have texture bias, especially when they are trained under ImageNet. To reduce this texture bias, Geirhos et al. propose a manual injection of share awareness by style-transferring ImageNet dataset [3]. Hermann et al. (which shares the most similar motivation with our study) unveiled the origins of texture bias and the reason why CNNs are inherently biased to texture information in the ImageNet dataset; The data distribution of ImageNet causes the model to classify labels by texture characteristics, not shape information. Moreover, this study further claimed that simply elevating shape awareness does not always contribute to the best performance; thus, a careful approach to bias mitigation should be considered. Beyond the aforementioned analyses on ImageNet-trained CNNs, Islam et al. designed a method of quantitatively analyzing CNN's shape and texture biases[15].

3 Adaptive Bias Analysis (AdaBA)

3.1 Baseline and its Improvement Avenues

Before performing analyses, we strongly necessitate an effective, quantitative tool to understand the CNN's texture and shape bias in a more precise manner. While Islam et al. presented a solid baseline approach, we scrutinized several improvement avenues. The detailed limits of the baseline is described below. For descriptions on the baseline, please refer to the original publication [15], and we also provided brief illustrations in the supplementary materials.

Dependence on heuristically-defined texture patterns The baseline requires heuristically-defined texture patterns to create shape and texture pairs. We analyze it induces the proposed method to exhibit CNN's bias focused on this heuristically-chosen texture, not the original texture information included in the original sample. If we desire to claim that 'CNN trained on dataset X is biased to textures', defining the shape and texture pairs should not rely on heuristically-chosen texture patterns. Instead, we presume an improved approach should be capable of establishing shape and texture pairs without any human interventions.

Training additional style transfer model Furthermore, the baseline also requires establishing an additional style transfer method to create texture and shape pairs. Then, it implies that we should additionally train an auxiliary model to utilize the baseline. We further expect this point as a risky factor of the baseline. What if the style transfer model is not qualified enough to provide texture-invariant samples?

Misled interpretation on bias score derived from Softmax Function Last but not least, we figured out the case where the final bias scores are not consistently sustained with the calculated mutual information which implicitly describes the strength of bias. Referring to the baseline's score calculation procedures, it applies a softmax operation to the set of mutual information from texture pair, shape pair, and residuals. For example, suppose the case where the set of mutual information is [-0.9959, 0.8498, 1] for texture, shape, and residuals, respectively. As mutual information exhibits similar implications to the correlation coefficient, the aforementioned set shows a strong texture bias rather than shape. But, when we apply a softmax operation to this set, a mutual information score for texture becomes 0.22 while this score for shape becomes 0.77, exhibiting high shape bias. We analyze the softmax operation as a risk of reversing the original mutual information on texture and shape information, which might cause a misinterpretation of CNN's bias.

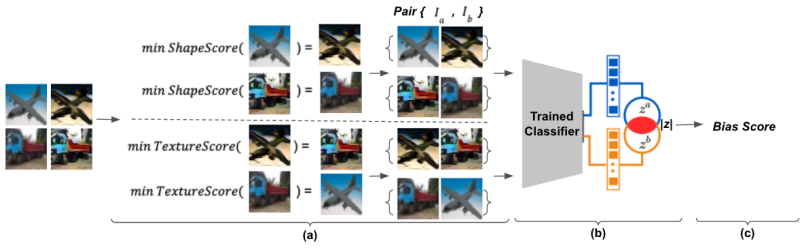


Fig. 1. Illustration of AdaBA. (a) defining texture and shape pair with shape and texture score respectively, (b) extracting mutual information from defined pair, (c) we use raw mutual information with absolute value. Note that mutual information of (b) defined in Islam et al. [15]

3.2 Methodology

Considering the baseline's improvement avenues, we design a novel approach to analyze CNN's bias in a quantitative manner. We denote our approach as the Adaptive Bias Analysis (AdaBA), as the proposed method can yield shape and texture bias scores adaptive to the dataset where the CNN is trained. We visualized an overall architecture of AdaBA in Figure 1. The proposed AdaBA resolves the aforementioned drawbacks by changing baseline approach's two components:

1) Texture and shape pair generation without auxiliary components, 2) Redesigning bias score by eliminating softmax operation. The detailed descriptions are described below.

Texture and shape pair generation without auxiliary components To resolve the first and second drawbacks of the baseline, AdaBA creates shape and texture pairs leveraging the theoretical definition of texture and shape proposed in style transfer studies. Referring to the design philosophy of the baseline, the texture pair includes two samples that have similar textures and different shapes. For a shape pair, vice versa. The pair generation procedures of AdaBA are as follows. First, it extracts the feature map from CNN on a given image sampled from the target dataset. We denote F^a as a feature map of the image I_a. Second, it generates two matrices of shape matrix and gram matrix, which are proposed in style-transfer studies [6]. A shape matrix can be defined as F_{ij}, and the texture(gram) matrix is justified as $\sum_k F^a_{ik} F^b_{jk}$ where i, j indicates width and heights at the feature map. Note that gram matrix implicit the meaning of texture information in a given image [6]. Third, for every sample in the dataset, it calculates the shape and gram matrix. It then selects a single sample as an anchor and calculates both shape score and texture score with the other samples following the equations provided in 1 and 2, respectively (i.e., measures the shape and texture scores from a pair consists of one anchor sample and one the other sample). The shape score exhibits a euclidean distance of shape matrices over the euclidean distance between gram matrices. The large shape score means a large shape difference over texture difference; thus, it implies two samples in a pair have dissimilar shape characteristics over the texture information. For the texture score, vice versa. Note that the texture score is a reciprocal of the shape score. Lastly, given an anchor sample, we establish a shape pair by selecting another sample that records the lowest shape score. For a texture sample, the texture pair is established with another sample with the lowest texture score. Throughout these procedures, key benefits of AdaBA's pair generation are particularly vivid. It does not require heuristically-chosen texture patterns or an additional style-transferring model, while it establishes shape and texture pairs based on a solid theoretical definition of texture and shape.

$$Shape\ Score = \frac{Euclidean(F^a_{ij}, F^b_{ij})}{Euclidean(\sum_k F^a_{ik} F^b_{jk}, \sum_k F^a_{ik} F^b_{jk})} \tag{1}$$

$$Texture\ Score = \frac{Euclidean(\sum_k F^a_{ik} F^b_{jk}, \sum_k F^a_{ik} F^b_{jk})}{Euclidean(F^a_{ij}, F^b_{ij})} \tag{2}$$

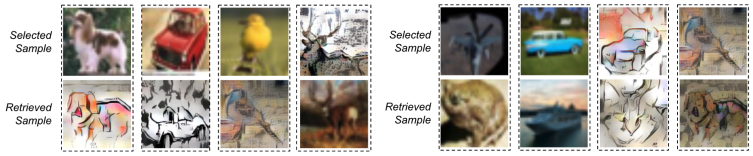
Redesigning bias score by eliminating softmax operation AdaBA employs the absolute value of calculated mutual information on shape and textures instead of softmax operation. As we discovered the use of softmax operation risks misleading CNN's bias, we designed AdaBA to eliminate its use. By eliminating

the use of softmax operation on MIs, we presume the risk of bias misinterpretation decreases compared to the baseline approach. Given mutual information values on texture and shape, our study used an absolute value of them as the final texture and shape bias score. We presume the mutual information values resulting from the CNN include the original representation of the biases without much noise; thus, retrieving an absolute value of each mutual information better describes the CNN's biases.

3.3 Validation on AdaBA

To prove a usefulness of the proposed AdaBA, we establish two questions for examining AdaBA's validity: 1) whether the AdaBA generates shape and texture pairs well, 2) whether the AdaBA yields bias analysis results on ImageNet-trained CNNs consistent with previously-proven discoveries. Upon the proven validity of AdaBA, we further scrutinize the behaviors of fine-tuned CNNs, which is absent in prior works.

Does AdaBA generate the pair well? First, as our AdaBA aims to alternate the baseline's pair generation procedure, we hereby examine whether the established shape and texture pairs well consistent with the original purpose; shape pairs should include samples that share a similar shape and dissimilar texture, while texture pairs include samples that have dissimilar shapes and similar texture. Given a concatenated dataset that includes both original CIFAR-10 and Stylized CIFAR-10, we sampled several shapes and texture pairs (which are established through AdaBA) and visualized them in Figure 2. Following the established pairs, we qualitatively evaluate that the generated shape and texture pair satisfy their original purpose; therefore, the proposed AdaBA effectively creates those pairs without heuristically-defined patterns and auxiliary style-transfer model.



(a) Retrieved shape Pairs (b) Retrieved texture Pairs

Fig. 2. Generated shape and texture pairs with AdaBA. We observe that generated shape and texture pairs fully satisfy the original purpose of pair generation.

Does AdaBA yield consistent results with prior discoveries? Furthermore, our study also aims to evaluate whether the proposed AdaBA effectively describes CNN's shape and texture bias. We actively referred to their evaluation logic of the baseline study, which is a state-of-the-art. The baseline study

evaluated their approach's performance by comparing the resulted shape and texture bias scores with prior studies' discoveries. For example, various studies suggested that ImageNet-trained CNNs are biased toward image textures rather than shapes. The baseline approach also yields a higher texture bias score than the shape score; thus, the prior studies and the baseline approach are saying the same proposition, so this study urges that their approach is correct. Please refer to the original publication [15] for more detailed explanations. Upon this evaluation logic, we examine whether the AdaBA yields bias analysis results consistent with the following discoveries [15]: *1) ImageNet-trained CNN is biased to the texture*, *2) Style-transferring reduces this texture bias.* For the dataset in these examinations, we alternatively used TinyImageNet (a subsample of ImageNet) as analyses on the original ImageNet required large computation overheads.

To implement a CNN trained under TinyImageNet, we trained the ResNet-50 model with the TinyImageNet dataset. With these trained CNNs, we compared the bias analysis results at AdaBA and the baseline to prove whether our approach yields similar analysis results consistent with the previous well-proven discoveries. Note that we also showed the baseline method's analysis results as it is known to show similar results to previous discoveries. The results are shown in Table 1.

Table 1. Bias analysis on TinyImageNet

Dataset	AdaBA			Baseline		
	Shape	**Texture**	$\frac{Shape}{Texture}$	**Shape**	**Texture**	$\frac{Shape}{Texture}$
Original	0.2202	0.3756	0.5863	518	1530	0.3386
Stylized	0.3842	0.3295	1.1660	755	1293	0.5839

From the results, we discovered that AdaBA consistently accomplishes similar results to the previous discoveries. First, we figure out that both AdaBA and baseline method yields texture bias of TinyImageNet-trained CNN (which is a proxy of ImageNet-trained one), where this texture bias has been proved in previous works. Second, we also discovered both approaches yield that the CNNs trained under the style-transferred samples exhibit a reduced texture bias rather than the one trained with original samples. The AdaBA and baseline method exhibits an increased $\frac{Shape}{Texture}$ value, where the larger value implies enhanced shape bias and reduced texture bias. Throughout these results, we conclude that the proposed AdaBA exhibits bias analysis results consistent with the baselines while it improves the baseline's limits simultaneously. We acknowledge that the proposed AdaBA should be examined in various datasets or problem settings but skipped in this work as our study primarily focuses on the proposing solution to mitigate texture bias, not analyzing the bias.

3.4 Bias Analysis on Fine-tuned CNNs with AdaBA

Upon the theoretical and experimental justification of our AdaBA, we extend our focus toward the fine-tuned CNNs. As no analyses exist on the fine-tuned CNNs' biases, we hereby scrutinize them. We postulate two questions regarding fine-tuned CNN's bias as follows: *1) What bias do CNNs fine-tuned in various benchmark datasets expose?* Do they show texture bias just as it showed in ImageNet? *2) Does style transferring indeed mitigate texture bias in the fine-tuned CNNs, just as it does in ImageNet?* While prior study [7] suggested that

style-transferring contributes to mitigating texture bias, is this method also valid in the fine-tuning regime? For implementation details, we employ four conventionally-utilized benchmark datasets: CIFAR-10, CIFAR-100, TinyImageNet, and Stanford Cars. Given these datasets, we fine-tuned the ResNet-50 classifier from the ImageNet-trained weights and retrieved shape and texture scores from our AdaBA and the baseline method. To answer the first question, we examine whether the fine-tuned CNNs are also implicit texture bias, as in ImageNet-trained CNN. Moreover, for the second question, we style-transfer CIFAR-10 with AdaIN and fine-tune the CNN on this style-transferred dataset. We then compare its texture and shape bias with the one fine-tuned in the original CIFAR-10 dataset. Throughout these setups, the experiment results are shown in Table 2 and Table 3, and key findings are illustrated as follows.

Table 2. Bias analysis on Fine-tuned CNNs. AdaBA results in similar trends to the baseline method. Note that MI means mutual information.

Dataset	AdaBA				Baseline			
	ShapeMI	TextureMI	Shape	Texture	ShapeMI	TextureMI	Shape	Texture
CIFAR-10	-0.2170	0.7271	0.2170	0.7271	-0.0994	-0.1039	1026	1022
CIFAR-100	-0.1631	0.2174	0.1631	0.2174	-0.4479	0.4659	586	1462
TinyImageNet	-0.2202	0.3756	0.2202	0.3756	-0.4731	0.5674	534	1514
Stanford-Cars	-0.0045	0.7894	0.0045	0.7894	0.4339	-0.5646	1496	552

Fine-tuned CNNs exhibit texture bias, but their strengths depend on the downstream dataset We scrutinize that fine-tuned CNNs also bear texture bias just as the one trained in ImageNet; thus, the analysis proposed in Geirhos et al. [7] is also valid in the fine-tuning scenario. Furthermore, we hypothesize a recent study of Hermann et al.[14] on the origins of CNN's texture preferences also supports this result, implying that both ImageNet and widely-used benchmark datasets' data distribution become a root cause of texture bias. As Hermann et al.[14] once noted, we also suspect that the model architecture is not a big concern, but the data distribution or label space design provokes this bias.

Style-transferring do mitigate fine-tuned CNN's texture bias Following Table 3, we observe that style-transferring the downstream dataset contributes to mitigating the texture bias of the fine-tuned CNN. Just as the prior study pointed out, we presume style-transferring samples unify the texture information in the dataset, thus enhancing the awareness of shape information to the CNN. As a comparative strength of shape over the texture increases after style transfer, we result that the method proposed in [7] is also valid under the fine-tuning regime.

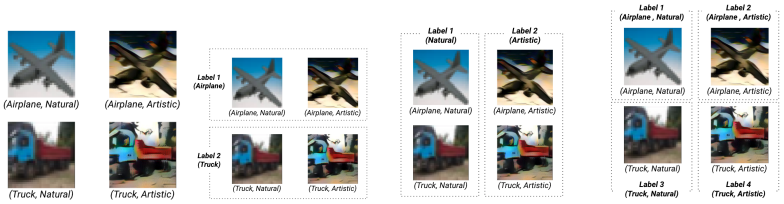
Table 3. Bias analysis on CIFAR-10

Dataset	AdaBA (Ours)			Baseline		
	Shape	Texture	$\frac{Shape}{Texture}$	Shape	Texture	$\frac{Shape}{Texture}$
Original	0.2170	0.7217	0.2984	180	1259	0.1430
Stylized	0.2882	0.1576	1.8286	330	516	0.6395

4 Mitigating Texture Bias by Redesigning Label Space

4.1 Granular Labeling Scheme

Objective Upon the takeaway that data distribution is one promising cause of CNN's texture bias, then, how can we figure out the optimal balance between texture and shape bias? Prior works primarily focused on mitigating texture bias under the frozen label space, which implies that a training set was regarded as a sanctuary. But, our study ideates that a practical solution should release this assumption and changes the data distribution by redesigning the label space. Accordingly, we suggest a novel labeling scheme denoted as a Granular labeling scheme where the samples belonging to each label share the shape and texture characteristics simultaneously, while conventional label spaces are established based on the human practitioner's needs.



(a) Unlabeled (b) Shape-biased (c) Texture-biased (d) Granular (Ours)

Fig. 3. Various labeling schemes that we utilized in the study. Given unlabeled samples shown in (a), the machine learning practitioners can create various labeling schemes: shape-biased, texture-biased, and granular schemes.

Dataset As we aim to examine the effectiveness of various labeling schemes, we necessitate a set of unlabeled samples that can be annotated under various labeling schemes. To the best of our knowledge, we could not figure out publicized datasets fulfilling this requirement; thus, we synthetically created the dataset. Given CIFAR-10, first, we style-transferred the original CIFAR-10 where its samples bear similar shapes and dissimilar textures from the original CIFAR-10. As we showed in Figure 3, our study concatenated them with the original CIFAR-10 and regarded it as a set of unlabeled samples; thus, the concatenated samples have ten shapes (airplane, automobile, $\cdots$), and two textures (Natural and Artistic). Given these concatenated samples, we created different training and test sets based on various labeling schemes. We denote a concatenated dataset consisting of original CIFAR-10 with three styles (Mosaic Realism, Rococo, and Neoplasticism) as Concatenated Set 1, 2, and 3, respectively.

Labeling Scheme We postulated three labeling schemes that the machine learning practitioners would utilize: Shape-biased, Texture-biased, and Granular labeling scheme (which is our proposition). Suppose the practitioners have a

set of unlabeled samples as shown in Figure 3. (a), where there exists two shape characteristics (airplane, truck) and two texture characteristics (natural, artistic). Under the shape-biased scheme shown in Figure 3. (b), the practitioners can establish two labels of airplane and truck, and samples within each label share similar shape characteristics but different textures (i.e., samples in label 1 have a similar airplane-like object, but their texture varies). Conversely, under the texture-biased scheme shown in Figure 3. (c), samples at each label share similar texture characteristics but dissimilar shapes (i.e., samples in label 1 share natural texture but have different objects). Lastly, the granular labeling scheme (a novel labeling scheme proposed in our study) lets the samples at each label be differentiated in both shape and texture characteristics; thus, the number of labels increases compared to the prior ones. Referring to Figure 3. (d), the samples in label 1 have different shape and texture characteristics from the other labels. Based on these schemes, we examined whether the proposed granular labeling scheme conveys better representation power to the model for various computer vision tasks.

Classification Setting and Evaluation Our study posits two classification tasks: shape classification and texture classification. Suppose we solve a binary classification between truck and airplane labels given a concatenated dataset (which simultaneously includes Natural and Shape texture) for shape classification. In this case, the practitioners conventionally design a labeling scheme where the samples at each class share similar objects and different textures. The samples in the truck label share look-a-like truck objects, but their textures vary from Natural to Artistic ones. Following the defined labeling schemes in section 4.1 (b), we can say the practitioners followed the shape-biased labeling scheme, and the model would solve binary classification between truck and airplane labels. As the granular scheme divides each label more finite manner, the model shall solve a 4-class classification with the following labels: (truck, natural), (truck, artistic), (airplane, natural), and (airplane, artistic). For a proper comparative evaluation, we concatenated the prediction results under the granular scheme based on the shape property. For example, we concatenated (truck, natural) and (truck, artistic) prediction results as a truck label, and the airplane label consists of prediction results of (airplane, natural) and (airplane, artistic). Therefore, the model under the granular scheme solves the 4-class classification, but the machine learning practitioners can practically acquire binary labels by concatenating prediction results fit to their classification objective. Last but not least, we utilized Accuracy and F1-score as evaluation metrics.

4.2 Bias Analysis on Concatenated Sets

In this section, we aimed to determine whether the strength of texture changes depending on the labeling scheme. Moreover, we further examine whether our granular scheme achieves a different bias level compared to the other schemes. Given three concatenated sets, we performed bias analysis with the proposed

AdaBA and measured the strength difference between texture and shape bias (denoted as *Diff*). Following the results of Table 4, we scrutinized that different labeling schemes exhibit different bias landscapes of the CNN; thus, redesigning the label space would presumably influence a balance between texture and shape biases. We further discovered that our proposed granular scheme achieves a balanced point between shape-biased and texture-biased schemes' one-side bias. This implies representation with conventional labeling schemes (i.e., shape-biased or texture-biased) overly biased to one side, but the proposed granular scheme can contribute to the balanced landscape of CNN's bias in texture and shape.

Table 4. Bias analysis on CNNs trained under various labeling schemes. Note that Diff implies an absolute value of difference between **Shape** and **Texture** scores.

Labeling Scheme	Training Sets								
	C-Set1			C-Set2			C-Set3		
	Shape	Texture	Diff	Shape	Texture	Diff	Shape	Texture	Diff
Granular	0.5588	0.2785	**0.2803**	0.2356	0.6270	**0.3914**	0.6227	0.8230	**0.2003**
Texture-biased	0.3526	0.4641	0.1115	0.0584	0.7532	0.6948	0.5330	0.6587	0.1258
Shape-biased	0.1698	0.9615	0.7917	0.9959	0.8498	0.1461	0.0461	0.9460	0.9000

5 Is Granular Labeling Scheme Advantageous in Classification Performance?

5.1 Setup

We first and foremost examined whether the proposed granular labeling scheme contributes to better performance at two classification tasks: shape classification and texture classification. The shape classification is a 10-class classification where the samples at each label include similar object shapes and different textures of natural and artistic. For the shape classification, we primarily compared the proposed granular scheme's performance with a shape-biased scheme. Note that the CNN trained under the granular scheme solves 20-class classification while the shape-biased scheme lets the model solve 10-class classification. On the other hand, the texture classification is a binary classification with two labels: natural and artistic. We validate the effectiveness of the granular scheme with the texture-biased scheme. While the granular scheme shares the same setting with the one at shape classification, the texture-biased scheme takes binary label space: the samples at each class share similar textures and different shapes. We followed the evaluation procedures described in section 4.1 for a proper comparative study. The experiment results are described in Table 5.

5.2 Analysis

We discovered the proposed granular labeling scheme contributed to precise classification performances compared to other paradigms. We hypothesize conveying both shape and texture characteristics contributed to more qualified representation, and this improved representation supported a significant classification

performance at both tasks. Throughout the experiment results, we reconfirmed a common notion: the samples within a single label shall share similar characteristics, and the more minimized variance contributes to the better representation power for classification. Accordingly, we figured out that a simple label space change can improve classification performance; thus, this finding can be a useful guideline for machine learning practitioners.

Table 5. Shape and texture classification results under various labeling schemes. Denote that C-set means concatenated sets.

Labeling Scheme	Training Sets					
	C-Set1		C-Set2		C-Set3	
Shape Classification						
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Shape-biased	0.7751	0.8625	0.7851	0.8679	0.6990	0.8153
Granular	**0.7835**	**0.8884**	**0.7858**	**0.8769**	**0.7337**	**0.8403**
Texture Classification						
	Accuracy	F1-score	Accuracy	F1-score	Accuracy	F1-score
Texture-biased	0.9611	0.9796	0.9621	0.9804	0.9837	0.9916
Granular	**0.9857**	**0.9905**	**0.9900**	**0.9938**	**0.9933**	**0.9950**

6 Does Granular Labeling Scheme Contribute to Better OOD Detection?

6.1 Setup

Furthermore, we validate whether the proposed granular labeling scheme contributes to better OOD detection performance. Among previous OOD detection methods [12,13,19], we employed an approach proposed in Vaze et al.[26] due to its supreme performance in various benchmark datasets. As we trained the CNN with the dataset stemming from CIFAR-10, we utilized two OOD datasets that do not share the same semantics with the training set: CIFAR-100 and SVHN. Furthermore, we synthetically created additional OOD samples for a more precise experiment: a stylized OOD dataset. The stylized OOD dataset includes samples that have been style-transferred with the AdaIn under the same procedure of creating the training set. Note that we created style-transferred OOD samples using the same style type used in the training set. (i.e., If the training set includes stylized samples in Rococo type, the OOD dataset also includes stylized OOD samples with Rococo type) We established the OOD detectors based on three CNNs trained under different labeling schemes and examined which labeling scheme contributes to better OOD detection performance. Following prior OOD detection studies, we also employed Area Under ROC curve (AUROC) for an evaluation metric to comprehensively evaluate the OOD detection performance under various threshold levels. Note that we utilized the same implementation settings with the one elaborated on section 5. We described experiment results in Table 6.

6.2 Analogy

We figured out a model trained under the granular labeling scheme was not always superficial in detecting OOD samples. While the granular scheme ac-

complished precise OOD detection performance in the Stylized OOD dataset, the shape-biased scheme achieved better performance in most original OOD datasets. We presume an underlying reason for this result also lies in the overfitted representations under the granular labeling schemes. As the granular labeling scheme acquires overly optimized representations to the training samples, it weakens the general understanding of various samples, including the ones that exist at different distributions. Based on the results of the OOD detection, we expect the proposed granular scheme is vulnerable in vision tasks which include samples at different distributions. Still, we acknowledge our analogy is at an empirical level; thus, a more in-depth analysis of this phenomenon is highly required in the follow-up studies.

Table 6. The OOD detection performances under various labeling schemes. Denote that C-set means concatenated sets.

OOD Set	Labeling scheme	Original OOD			Style-Transfered OOD		
		C-Set1	C-Set2	C-Set3	C-Set1	C-Set2	C-Set3
CIFAR-100	Shape-biased	**0.7787**	**0.8068**	0.7023	0.7154	0.7391	0.7153
	Texture-biased	0.4604	0.5564	0.6598	0.7524	0.7317	0.6102
	OURS	0.6745	0.7651	**0.7942**	**0.8636**	**0.8696**	**0.7527**
SVHN	Shape-biased	**0.7734**	**0.7981**	0.6876	0.7756	0.5583	0.7458
	Texture-biased	0.4355	0.4575	0.4764	0.7503	0.5446	0.5841
	OURS	0.7187	0.7587	**0.6943**	**0.8892**	**0.5858**	**0.7746**

7 What Representation Do Various Schemes Acquire?

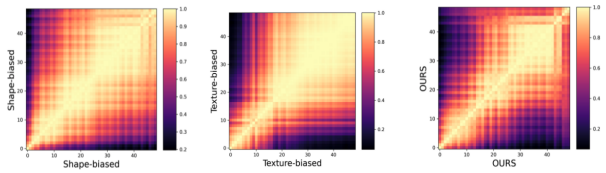
7.1 Setup

While we discover that the proposed granular scheme acquires a balance between texture and shape information, the follow-up question arises: *How does the learned representation look like?* To excavate an answer, we compared the representation similarities among CNNs trained under various labeling schemes with CKA [17]. We analyzed layer-wise representation similarity within a single model, that implies a similarity among convolution layers in a single CNN. We aim to analyze the knowledge capacity of a trained model. Suppose particular layers in a single model acquires representations similar to the other layers. This case implies that the model failed to learn various characteristics of the training samples; thus, it limits the representation power of a given sample. Conversely, if layers within the same model bear lower similarities to each other, it shows that the model can illustrate various characteristics of a given sample; thus, the model considers a wide range of patterns exists to solve a classification task. Upon the aforementioned setups, we visualized layer-wise representation similarities within each model Figure 4. Note that both x and y-axis in the figure imply the convolution layers index at ResNet-50.

7.2 Analogy

Following the results in Figure 4, we figured out the CNN trained under the granular scheme has a large capacity of knowledge as its layer-wise representation

similarity is comparatively lower than the others. While the CNNs trained under shape-biased and texture-biased schemes bear many similar representations within their layers, the model trained under the proposed scheme has smaller similar representations. We analyze this smaller similarity among layers let the model scrutinize various patterns of a given sample, and this larger knowledge capacity contributed to the precise classification performance. We figured out the effectiveness of the proposed granular labeling scheme comes from the quality of representation. The representation trained under the proposed scheme has a larger knowledge capacity, and it acquires a presumably qualified contextual understanding of a given data at high-level layers of the neural networks. For more in-depth analyses, we additionally revealed that a CNN trained under the granular labeling scheme exhibits distinct high-level representations from the other schemes; thus, this distinct representation particularly contributes to better performances. We skipped the description in this paper due to page limits, please refer the supplementary materials for detailed analyses.



(a) Shape-biased (b) Texture-biased (c) Granular (Ours)

Fig. 4. Layer-wise representation similarities among layers within the same model.

8 Conclusion

Throughout the study, we present a series of analyses that scrutinize CNN's texture and shape bias. First, we propose AdaBA, a novel bias analysis method that sustains the baseline method's performance as well as more lightweight procedures. Upon the AdaBA, we explore how the fine-tuned CNNs expose a biased landscape in various downstream datasets and result in a data distribution is a root cause of texture bias of fine-tuned CNNs, as well as ImageNet-trained CNNs [14]. To this end, we suggest a granular labeling scheme that can mitigate the CNN's texture preference by simply redesigning the label space. We empirically examine that the granular labeling scheme exhibits a balanced bias between texture and shape, and it yields escalated performances on classification and OOD detection. Lastly, we analyze that the granular labeling scheme acquires more qualified representation power, describing a fruitful illustration of a given sample.

Acknowledgement This work is supported by the Korea Agency for Infrastructure Technology Advancement grant funded by the Ministry of Land, Infrastructure, and Transport (Grant RS-2022-0014579).

References

1. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps. Advances in neural information processing systems **31** (2018)
2. Cheng, B., Wei, Y., Shi, H., Feris, R., Xiong, J., Huang, T.: Revisiting rcnn: On awakening the classification power of faster rcnn. In: Proceedings of the European conference on computer vision (ECCV). pp. 453–468 (2018)
3. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. pp. 248–255. Ieee (2009)
4. Dodge, S., Karam, L.: A study and comparison of human and deep learning recognition performance under visual distortions. In: 2017 26th international conference on computer communication and networks (ICCCN). pp. 1–7. IEEE (2017)
5. Doshi-Velez, F., Kim, B.: Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608 (2017)
6. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2414–2423 (2016)
7. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. arXiv preprint arXiv:1811.12231 (2018)
8. Ghorbani, A., Wexler, J., Zou, J.Y., Kim, B.: Towards automatic concept-based explanations. Advances in Neural Information Processing Systems **32** (2019)
9. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J., et al.: Recent advances in convolutional neural networks. Pattern Recognition **77**, 354–377 (2018)
10. Guo, C., Wu, D.: Canonical correlation analysis (cca) based multi-view learning: An overview. arXiv preprint arXiv:1907.01693 (2019)
11. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
12. Hendrycks, D., Gimpel, K.: A baseline for detecting misclassified and out-of-distribution examples in neural networks. arXiv preprint arXiv:1610.02136 (2016)
13. Hendrycks, D., Mazeika, M., Dietterich, T.: Deep anomaly detection with outlier exposure. arXiv preprint arXiv:1812.04606 (2018)
14. Hermann, K., Chen, T., Kornblith, S.: The origins and prevalence of texture bias in convolutional neural networks. Advances in Neural Information Processing Systems **33**, 19000–19015 (2020)
15. Islam, M.A., Kowal, M., Esser, P., Jia, S., Ommer, B., Derpanis, K.G., Bruce, N.: Shape or texture: Understanding discriminative features in cnns. arXiv preprint arXiv:2101.11604 (2021)
16. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: International conference on machine learning. pp. 2668–2677. PMLR (2018)
17. Kornblith, S., Norouzi, M., Lee, H., Hinton, G.: Similarity of neural network representations revisited. In: International Conference on Machine Learning. pp. 3519–3529. PMLR (2019)

18. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
19. Lee, K., Lee, H., Lee, K., Shin, J.: Training confidence-calibrated classifiers for detecting out-of-distribution samples. arXiv preprint arXiv:1711.09325 (2017)
20. Li, J., Zhu, G., Hua, C., Feng, M., Li, P., Lu, X., Song, J., Shen, P., Xu, X., Mei, L., et al.: A systematic collection of medical image datasets for deep learning. arXiv preprint arXiv:2106.12864 (2021)
21. Papageorgiou, C., Poggio, T.: A trainable system for object detection. International journal of computer vision **38**(1), 15–33 (2000)
22. Shen, D., Wu, G., Suk, H.I.: Deep learning in medical image analysis. Annual review of biomedical engineering **19**, 221–248 (2017)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014)
24. Sun, Q.S., Zeng, S.G., Liu, Y., Heng, P.A., Xia, D.S.: A new method of feature fusion and its application in image recognition. Pattern Recognition **38**(12), 2437–2448 (2005)
25. Tuli, S., Dasgupta, I., Grant, E., Griffiths, T.L.: Are convolutional neural networks or transformers more like human vision? arXiv preprint arXiv:2105.07197 (2021)
26. Vaze, S., Han, K., Vedaldi, A., Zisserman, A.: Open-set recognition: A good closed-set classifier is all you need. arXiv preprint arXiv:2110.06207 (2021)
27. Wu, Z., Shen, C., Van Den Hengel, A.: Wider or deeper: Revisiting the resnet model for visual recognition. Pattern Recognition **90**, 119–133 (2019)