Supplementary material: What do neural networks learn in image classification? A frequency shortcut perspective

Shunxin Wang Raymond Veldhuis Christoph Brune Nicola Strisciuglio
University of Twente, The Netherlands

1. Datasets

1.1. Synthetic datasets

The frequency bands of each class in synthetic datasets are shown in Table 1. For each dataset, class C_3 has a bias to a specific band, and classes C_0 and C_1 are designed to contain frequencies from the other three bands. Class C_2 contains frequencies across the whole spectrum. Example images from the four synthetic datasets that we created are shown in Fig. 1. We designed the classes so that they have specific frequency characteristics. We induced different levels of class-wise difficulty when the NNs are trained to distinguish their samples. Across the four datasets, the images of class C_3 are easily distinguishable from those of the other three classes, as observed visually. This is because class C_3 has a frequency bias to a specific band, e.g. low-frequency bias in the Syn$_{B_1}$ dataset and high-frequency bias in the Syn$_{B_4}$ dataset. The images of classes C_0, C_1, and C_2 are visually similar across the four synthetic datasets. Despite the visual similarity, the images of class C_0 have *special patterns* consisting of a fixed set of frequencies across the spectrum. The *special patterns* are the designed characteristics making the images of class C_0 easily distinguishable from classes C_1 and C_2. Note that, the *special patterns* consist of eight frequencies that can be evenly filtered based on the band-stop filters we use during testing. This is to analyze how the NNs utilize frequency information from the *special patterns* fairly. The difference between classes C_1 and C_2 is the number of frequency bands sampled for the data generation. Class C_1 has one less sampling band than those of class C_2. However, for the images of classes C_1 and C_2, it is hard for human observers to identify their difference visually while NNs can, according to their classification results. On the other hand, classes C_0 and C_3 are easier for human observers to be visually distinguished.

1.2. OOD test data: ImageNet-SCT

ImageNet-SCT is specifically designed to validate the influence of frequency shortcuts on an unseen dataset, for models trained on ImageNet10. The analysis demonstrates that NNs might learn frequency shortcuts for easier classification, which correspond to texture-based or shape-based patterns. The classification dependency on the patterns shows that NNs might ignore other useful semantics. Therefore, to validate how this learning behavior affects OOD

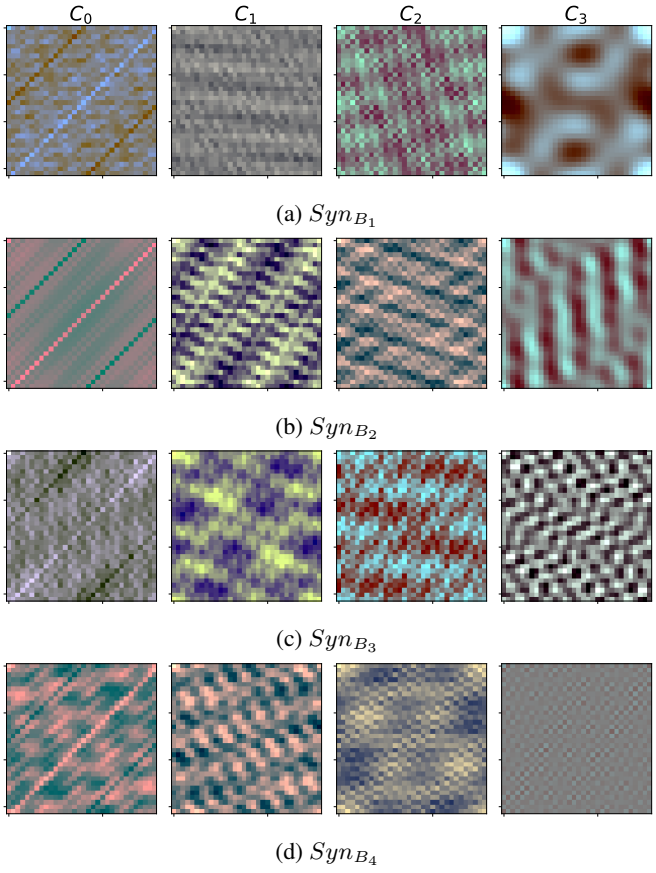


Figure 1: Samples of synthetic datasets. Class C_3 has a frequency bias to a specific band, which is B_1 for Syn_{B_1}, B_2 for Syn_{B_2}, B_3 for Syn_{B_3}, and B_4 for Syn_{B_4}. Due to frequency bias, images of class C_3 can be easily distinguished from other classes.

Table 1: Frequency bands of each synthetic dataset.

	Synthetic datasets			
class	Syn_{B_1}	Syn_{B_2}	Syn_{B_3}	Syn_{B_4}
C_0	B_2, B_3, B_4	B_1, B_3, B_4	B_1, B_2, B_4	B_1, B_2, B_3
C_1	B_2, B_3, B_4	B_1, B_3, B_4	B_1, B_2, B_4	B_1, B_2, B_3
C_2	B_1, B_2, B_3, B_4	B_1, B_2, B_3, B_4	B_1, B_2, B_3, B_4	B_1, B_2, B_3, B_4
C_3	B_1	B_2	B_3	B_4

generalization, we construct a new dataset, containing 10 classes similar to those of ImageNet-10 [4] but with different shape/texture characteristics. ImageNet-trained NNs are found to have a texture bias [1]. Thus, the main criterion applied for the composition of the dataset is to have classes with similar shape characteristics to ImageNet-10, instead of texture characteristics, except for classes 'military aircraft', 'car', and 'fishing vessel' which have similar texture characteristics to the corresponding classes in ImageNet-10. This helps to evaluate the influence of learned frequency shortcuts on an OOD test from two perspectives, namely when the shortcut features are present or absent. Each class contains 7 renditions of images (i.e. art, cartoon, deviantart, painting, sculpture, sketch and toy), which is inspired by the design idea of ImageNet-R [3]. Example images of ImageNet-SCT are shown in Fig. 2. Each row shows the images of the seven renditions of one class.

2. Training setup

Synthetic datasets. We train AlexNet [5], ResNet(s) [2] and VGG-16 [6] models for 100 epochs on the four synthetic datasets. The initial learning rate is 0.01, reduced by a factor of 10 if the validation loss does not decrease for 10 epochs. We use SGD optimizer with momentum 0.9 and weight decay 10^{-4}, and batch size 128.

ImageNet-10 dataset. Models with ResNet(s) [2] and VGG-16 [6] architectures are trained for 200 epochs on the ImageNet-10 dataset. The initial learning rate is 0.01 and is reduced by a factor of 10 if the validation loss does not decrease for 10 epochs. We use SGD optimizer with momentum 0.9 and weight decay 10^{-4}, and batch size 16.

3. Extra results

3.1. Synthetic datasets.

F_1-scores Fig. 3 shows the F_1-score computed on the test sets of the four synthetic datasets during the first 500 iterations of the training of AlexNet, ResNet9 and VGG16. As generally observable, all model architectures achieve higher F_1-scores for class C_3 than for the other classes. This indicates that class C_3 is recognized immediately and easily by the NNs during training. This is consistent with the results of ResNet18 and shows the existence of shortcut learning, which prioritizes the recognition of easily distinguishable frequency patterns.

Relative confusion matrices. Fig. 4 shows the relative confusion matrices of AlexNet (first column), ResNet9 (second column) and VGG16 (third column) trained on the four synthetic datasets. The models are tested on the different band-stop test sets, obtained by suppressing in turn the frequencies in two out of the four sub-bands considered for the data generation. Because of the class-wise frequency characteristics of the synthetic datasets, these tests are meant to inspect the frequency utilization of different NN models, i.e. what frequencies are needed for classification. The performance results of the models are mostly stable when they are tested on test sets retaining only two frequency bands (see the values of Δ^{C_i,C_i} where $C_i \in \{C_0, C_1, C_2, C_3\}$), showing that they do not need complete frequency information for classification. For instance, class C_0 has a *special pattern* consisting of frequencies across the whole spectrum, and the corresponding Δ^{C_0,C_0} is mostly close to zero. Models may find shortcut solutions in the Fourier domain for classification and this behavior is common across different architectures.

ADCS. The ADCS of classes in the four synthetic datasets are shown in Fig. 5. Across the four datasets, class C_3 has a significant bias on a specific band, from low to high. The yellow dots in $ADCS^{C_3}$ (belong to some frequencies in the frequency set of the *special pattern*) indicate that the corresponding frequencies have slightly more energy than other classes, which is caused by the removal of the specific frequencies (non-ideal filtering). Class C_0 has more energy around the specific frequency sets than other classes, this is also due to the non-ideal filtering. In general, the ADCS shows that the classes in a synthetic dataset Syn_b have distinguishable frequency characteristics. These might be used as shortcuts. The class with the most distinctive frequency characteristics, i.e. class C_3 is learned first by NNs in the training phase (see Fig. 3), indicating that the models have a tendency to identify that distinctive frequency characteristic as an easy solution for the classifica-

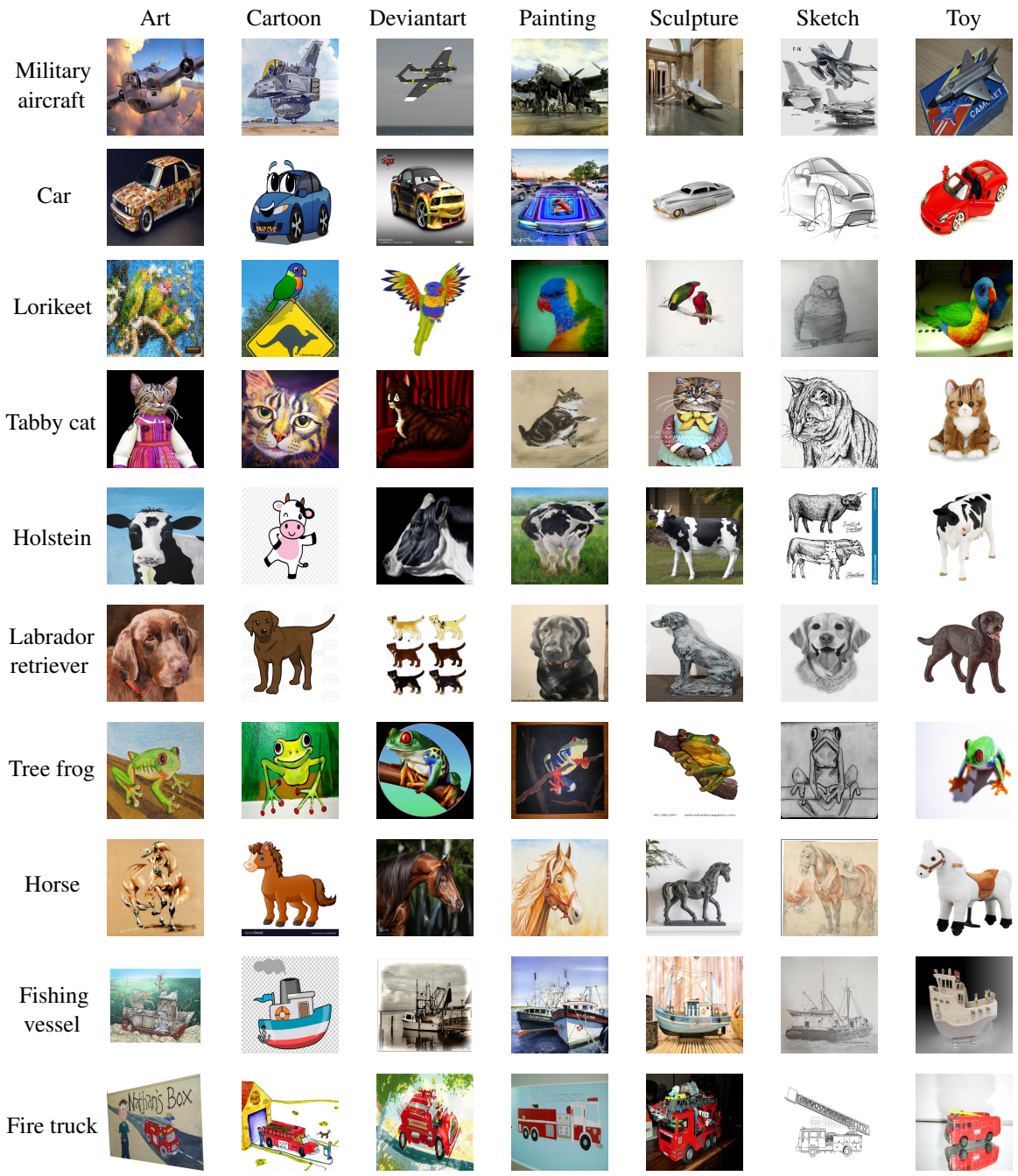


Figure 2: Example images from the ImageNet-SCT dataset. Images are organized in 10 classes, with images of seven different renditions: (in order of the columns) art, cartoon, deviantart, painting, sculpture, sketch, and toy.

tion problem. ADCS can be used to analyze the class-wise frequency characteristics in a dataset, rather than being used directly to predict which class might be learned first. Further investigation on frequency characteristics and learning dynamics is needed to establish if certain frequency characteristics induce a shortcut or not.

3.2. ImageNet-10

ADCS. The ADCS of other classes in ImageNet-10 are shown in Fig. 6. The classes have different frequency characteristics, which might be applied as discriminative features by NNs for classification. For instance, class 'siamese

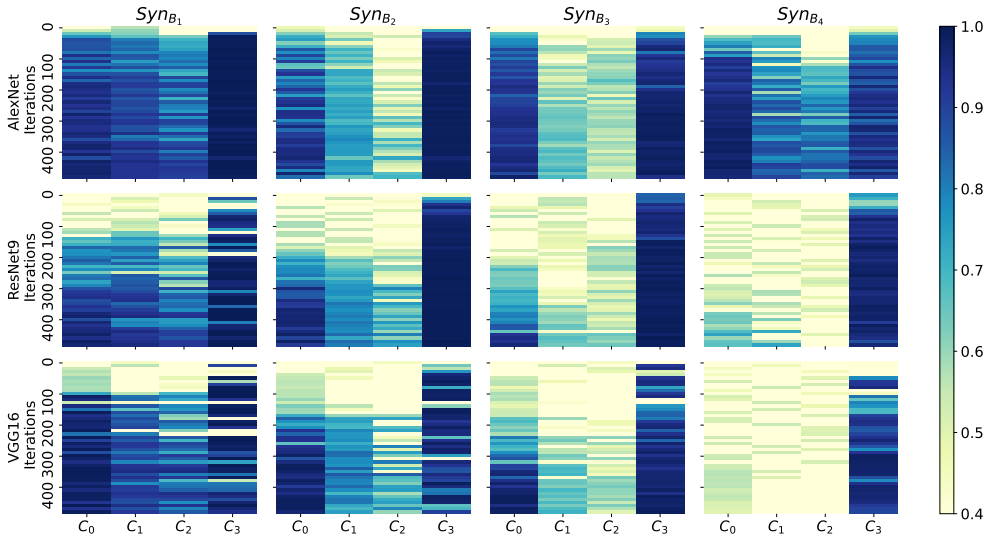


Figure 3: F_1-scores of the first 500 iterations of AlexNet, ResNet9, and VGG16 trained on the four synthetic datasets respectively.

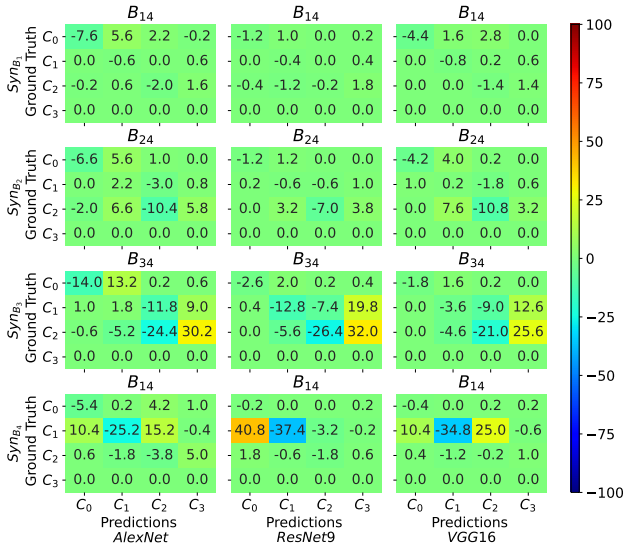


Figure 4: Relative confusion matrices of AlexNet, ResNet9, and VGG16 trained on the synthetic datasets

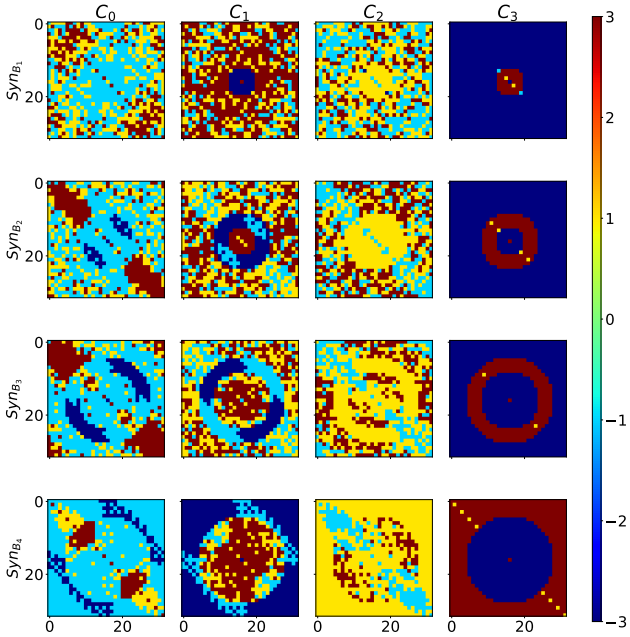
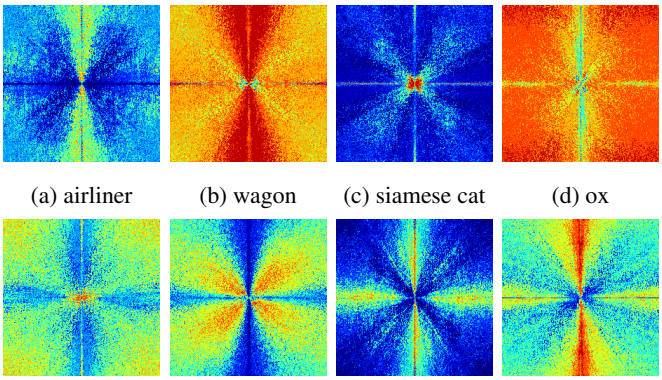


Figure 5: ADCS of the classes in synthetic datasets.

cat' has more energy in low-frequency compared to other classes, which is in line with the observation that the models use more low frequencies to classify the samples of 'siamese cat' from the top-5% DFM. Further, when using SIN (replacing textures while emphasizing shapes) to augment training data, ResNet18 learns a shape-bias frequency shortcut for it, showing the importance to analyze class-wise frequency characteristics of training data in image classification. Differently, the class 'container ship' has more energy on the frequencies whose spatial representations are horizontal and vertical lines. The ADCS of class 'trailer truck' shares similar characteristics to that of class 'container ship', but it does not have extremely low energy on high-frequency. Similar to the ADCS of class 'humming bird', class 'ox' has high energy in many high frequencies, though not as high as that of 'humming bird'. For other classes without obvious frequency differences, it is difficult to interpret the frequency utilization of the NNs, and thus



(a) airliner (b) wagon (c) siamese cat (d) ox

(e) golden retriever (f) tailed frog (g) container ship (h) trailer truck

Figure 6: ADCS of other classes in ImageNet-10.

we compute their DFMs.

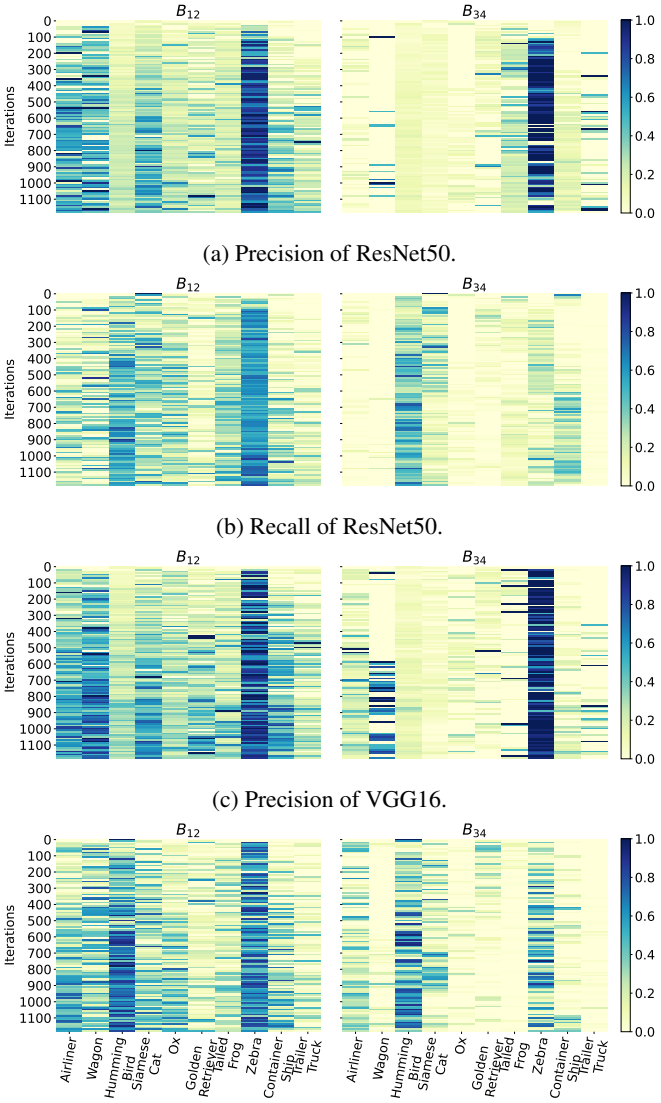
Precision and recall. We show the precision and recall of ResNet50 and VGG16 computed on the low-passed and high-passed test sets of ImageNet-10 (not the original test set), during the first 1200 iterations of training in Fig. 7. The models achieve generally higher precision and recall in the classes 'humming bird' and 'zebra'. This indicates that these classes have special characteristics that are easily used for classification by the models at the early training stages. The observations are in line with the learning behavior of ResNet18 trained on ImageNet-10 that we highlighted in the main paper, confirming that the bias of classification models is indeed driven by data characteristics, being low- or high-frequency components in the images according to the simplicity to solve the optimization problem.

Top-1% and top-10% DFMs. We show the top-1% and top-10% DFMs of each class for models trained on ImageNet-10 in Figs. 8a and 8b. We observe from the top-1% DFMs that NNs take the frequencies whose spatial representations are horizontal and vertical lines as the most dominant frequencies since the removal of them results in high loss increment. From the top-10% DFM in Fig. 8b, we observe the frequency utilization of NNs varies slightly across different architectures but shares similar patterns.

Results on ImageNet-10 DFM-filtered versions The classification results of models tested on ImageNet-10 DFM-filtered versions, with only the top-1% and top-10% dominant frequencies retained, are shown in Tables 2 and 3.

If a model uses 1% of frequencies and can achieve correct classification for most of the test samples, then it may not extract deep semantic information from the data and be subject to a shortcut learned during training. From Table 2, we observe that using only 1% of frequencies, ResNet18+AugMix predicts correctly 94% of the samples of class 'container ship' with FPR = 0.64, indicating a learned frequency shortcut and a strong bias towards a small set of frequencies. Interestingly, we observe VGG16, using only 1% of frequencies, learns a frequency shortcut for class 'ox', which has TPR = 0.76 and FPR = 0.35. ResNet18+SIN uses frequency shortcuts for classes 'siamese cat', 'ox', and 'golden retriever', observed from the high values of TPR and FPR.

By increasing the number of dominant frequencies considered in the input test images, as expected, all models achieve generally better performance for most of the



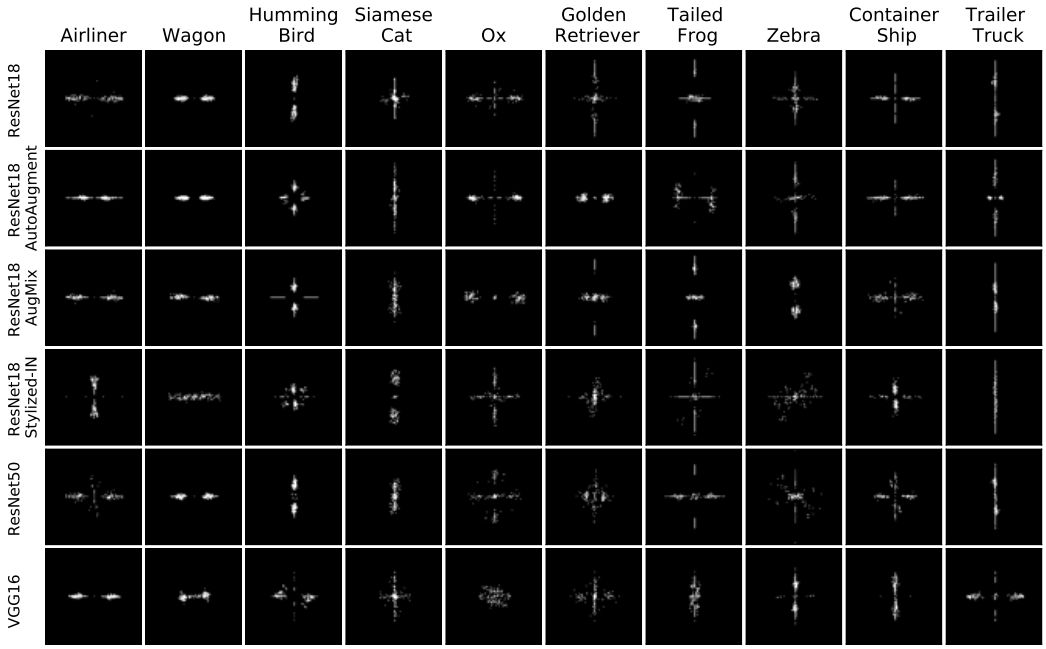
(a) Precision of ResNet50.

(b) Recall of ResNet50.

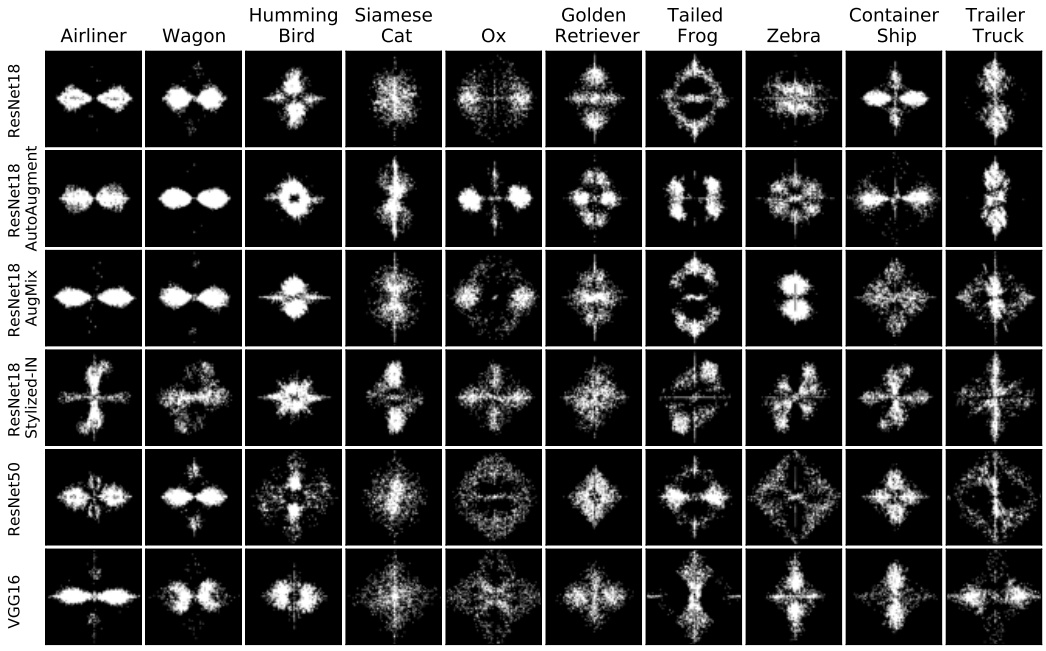
(c) Precision of VGG16.

(d) Recall of VGG16.

Figure 7: Precision and recall rates of ResNet50 and VGG16 trained on ImageNet-10 for the first 1200 iterations.



(a) Top-1%



(b) Top-10%

Figure 8: Dominant frequency maps of ResNet18 (with AutoAugment/AugMix), ResNet50 and VGG16. The maps show the (a) top-1% and (b) top-10% dominant frequencies of each class in ImageNet-10.

classes, compared to that on top-1% DFM-filtered test sets. From the results of models using the top-10% dominant frequencies for classification, we can, however, identify similar frequency shortcuts (to the identified frequency shortcuts using the top-5% dominant frequencies) from the Table 3. For instance, models other than ResNet18+AutoAug have high TPRs and FPRs for class 'container ship', indicating learned frequency shortcuts. For class 'zebra', ResNet18

Table 2: TPRs and FPRs on the top-1% DFM-filtered versions of ImageNet-10 (w/ df).

		ImageNet-10									
Model		**airliner**	**wagon**	**humming bird**	**siamese cat**	**ox**	**golden retriever**	**tailed frog**	**zebra**	**container ship**	**trailer truck**
ResNet18	TPR	0.08	0	0	**0.84**	0.02	0	0.24	0.38	**0.4**	0.24
	FPR	0.0067	0	0	**0.1133**	0.0156	0.0089	0.0133	0.0822	**0.1467**	0.0622
ResNet18+AutoAug	TPR	0	0	0	0.06	0	0	0	0.08	0.14	0.04
	FPR	0	0	0	0.0022	0	0	0.0022	0.0756	0.1067	0.0356
ResNet18+AugMix	TPR	0.02	0	0	0	**0.34**	0.1	0.06	0	**0.94**	0.1
	FPR	0.0044	0	0.0044	0	**0.2133**	0.0089	0	0	**0.64**	0.04
ResNet18+SIN	TPR	0.4	0	0.22	**0.88**	**0.74**	**0.72**	0	0	0	0.04
	FPR	0.1489	0.0044	0.2444	**0.42**	**0.4022**	**0.4889**	0.0022	0.0022	0.0044	0.0311
ResNet50	TPR	**0.34**	0	0	0.12	0	0.2	0	0.12	0.2	0
	FPR	**0.1**	0	0.0333	0.0133	0	0.04	0.0044	0.0489	0.0556	0.0111
VGG16	TPR	0.02	0	0	**0.64**	**0.76**	0.04	0.02	0.04	0.06	0.3
	FPR	0.0067	0	0	0.0978	**0.3556**	0.0311	0	0.0422	0.06	0.1422

Table 3: TPRs and FPRs on the top-10% DFM-filtered versions of ImageNet-10 (w/ df).

		ImageNet-10									
Model		**airliner**	**wagon**	**humming bird**	**siamese cat**	**ox**	**golden retriever**	**tailed frog**	**zebra**	**container ship**	**trailer truck**
ResNet18	TPR	0.2	0	0.62	0.92	0.06	0.16	0.12	**0.9**	**0.84**	0.02
	FPR	0.0067	0	0.0378	0.0556	0.0356	0.0156	0	**0.1156**	**0.2311**	0
ResNet18+AutoAug	TPR	0	0	0.22	**0.66**	0.2	0.18	0	**0.64**	0.02	0.02
	FPR	0	0	0.0067	**0.1267**	0.1067	0.0089	0	0.0289	0.0089	0.0022
ResNet18+AugMix	TPR	0.38	0	0.4	0.84	0.42	0.5	0.02	0.68	**0.9**	0.64
	FPR	0.0356	0	0.0089	0.06	0.1556	0.0156	0	0.0022	**0.1978**	0.0311
ResNet18+SIN	TPR	0.12	0.04	0.6	**0.88**	**0.94**	0.62	0.06	0.66	0.08	0.12
	FPR	0.0089	0.0067	0.02	**0.1044**	**0.3867**	0.0933	0.0022	0.0489	0.0667	0
ResNet50	TPR	0.44	0	0.04	0.72	0	0.42	0	0.12	**0.88**	0.1
	FPR	0.0733	0	0.0044	0.0378	0.0133	0.0311	0	0.04	**0.2356**	0.0178
VGG16	TPR	0.4	0	0.5	0.8	0.1	0.42	0.04	0.68	**0.82**	0.22
	FPR	0.0422	0	0.0311	0.0467	0.1133	0.0267	0	0.0378	**0.14**	0.0378

can predict 90% of the samples, with FPR = 0.1156, indicating another learned frequency shortcut. Moreover, ResNet18+SIN learns a frequency shortcut for class 'ox', while it is less biased to class 'siamese cat' with more frequencies provided (lower FPR compared to that of the model tested on the corresponding top-1% DFM-filtered test set). The identification of learned frequency shortcuts can be automatized by choosing the top-x% ranked frequency and setting thresholds (to the values of TPR and FPR) to evaluate the presence of shortcuts when testing the models on DFM-filtered test sets.

References

[1] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 2

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 2

[3] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8320–8329, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. 2

[4] Hanxun Huang, Xingjun Ma, Sarah Monazam Erfani, James Bailey, and Yisen Wang. Unlearnable examples: Making personal data unexploitable. In *International Conference on Learning Representations*, 2021. 2

[5] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 2

[6] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. 2