Robustmix: Improving Robustness by Regularizing the Frequency Bias of Deep Nets

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep networks have achieved impressive results on a range of well curated benchmark datasets. Surprisingly, their performance remains sensitive to perturbations that have little effect on human performance. We propose a novel extension of Mixup called Robustmix that regularizes networks to classify based on lower frequency spatial features. We show that this type of regularization improves robustness on a range of benchmarks such as Imagenet-C and Stylized Imagenet. It adds little computational overhead and furthermore does not require a priori knowledge of a large set of image transformations. We find that this approach further complements recent advances in model architecture and data augmentation attaining a state-of-the-art mean corruption error (mCE) of 44.8 with an EfficientNet-B8 model and RandAugment, which is a reduction of 16 mCE compared to the baseline.

1 Introduction

Deep neural networks have achieved state-of-the-art accuracy across a range of benchmark tasks such as image segmentation (Ren et al., 2015) and speech recognition (Hannun et al., 2014). However, their performance can suffer greatly in the presence of small data corruptions (Szegedy et al., 2014; Goodfellow et al., 2014; Moosavi-Dezfooli et al., 2017; Athalye et al., 2018; Hendrycks & Dietterich, 2018). Hendrycks & Dietterich (2018) show that the accuracy of a standard model on Imagenet can drop from 76% to 20% when evaluated on images corrupted with small visual transformations. Secondly, Szegedy et al. (2014) show the existence of *adversarial* perturbations which are imperceptible to humans but have a disproportionate effect on the predictions of a network.

Numerous proposals were made to improve the robustness of deep networks, including Hendrycks et al. (2019) which requires a priori knowledge of the visual transformations in the test domain. Others, such as Geirhos et al. (2018) use a deep network to generate transformations which comes with significant computation cost.

This paper proposes a new technique to improve the robustness of deep networks by regularizing their frequency bias. Its advantages are that: (1) it does not require knowledge of a large set of priori transformations, (2) it is inexpensive and (3) it doesn't have many hyper-parameters. The key idea is to bias the network to rely more on lower spatial frequencies to make predictions.

2 Related Work

The proposed approach can be seen as a generalization of Mixup (Zhang et al., 2018), a data augmentation method that regularizes models by training them on linear interpolations of two input examples and their respective labels.

Submitted to the Workshop on Distribution Shifts, 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

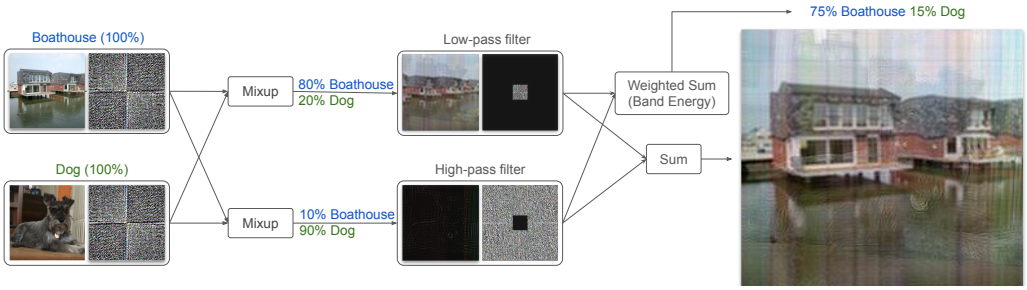


Figure 1: Illustration of the method. In order to better illustrate the method, we display the Fourier spectrum of the images next to them. We can see that even though 90% of the higher frequencies belong to the image of a dog, Robustmix assigns more weight to the boathouse label because it assigns more weight to the lower frequencies.

Zhang et al. (2018) show that Mixup improves the accuracy of networks and can also improve the robustness of the network. In the past years, several versions of Mixup were proposed with application in Fairness (Chuang & Mroueh, 2021), 3D reconstruction (Cheng et al., 2022), semi-supervised learning (Beckham et al., 2019), as well as robustness ((Mai et al., 2021; Yun et al., 2019; Faramarzi et al., 2020; Kim et al., 2020; Verma et al., 2019)). The novel version we propose here is frequency-based and does not include additional learnable parameters.

Augmix (Hendrycks et al., 2019) is a data augmentation technique to improve robustness by training on a mix of known image transformations. It adds little computational overhead, but requires knowledge of a diverse set of domain specific transformations. Hendrycks et al. (2019) mixes a set of 9 different augmentations to reach 68.4 mCE on Imagenet. In contrast, the proposed method does not rely on specific image augmentations but on the more general principle that natural images are a kind of signal where most of the energy is concentrated in the lower frequencies.

The idea of frequency filtering is popular in Deep learning frameworks and has numerous applications including unsupervised domain adaptation (Yang & Soatto (2020)) and adversarial perturbation attacks (Guo et al. (2018); Li et al. (2021)). Unlike the latter papers which focus on measuring the accuracy of a model after an adversarial attack, we focus on common (noise) corruptions by measuring mCE as robustness assessment.

Zhang (2019) uses low pass filters directly inside the model to improve the frequency response of the network.Wang et al. (2019) uses a differentiable neural network to extract textual information from images without modelling the lower-frequency. Our method also makes use of low-pass filtering but does not completely remove high frequency features. Additionally, we only use frequency filtering during training and therefore no computational overhead is incurred during evaluation.

3 Method

Motivation Wang et al. (2020) suggest that convolutional networks trade robustness for accuracy in their use of high frequency image features. Such features can be perturbed in ways that change the prediction of the model even though humans cannot perceive the change. This can lead models to make puzzling mistakes such as with adversarial examples. Our aim is to increase robustness while retaining accuracy by regularizing how high frequency information is used by the model.

Robustmix We propose to regularize the sensitivity of the model to each frequency band by extending Mixup's linear interpolations with a new type of band interpolation. The key insight is that we can condition the sensitivity to each band using images that mix the frequency bands of two different images. Suppose that we mix the lower frequency band of an image of a boathouse with the high frequency band of an image of a dog. We can encourage sensitivity to the lower band by training the model to predict dog for this mixed image. However,this approach is too simplistic because it completely disregards the impact of the image in the high band. Indeed, the ablation study at section A.1 shows us that it is not sufficient.

Instead, we interpolate the label of such mixed images according to an estimate of the importance of each frequency band. We propose to use the relative amount of energy in each band as an estimate

of the importance. Thus the sensitivity of the model to high frequency features will be proportional to their energy contribution in natural images. And as we can see in Figure 2a, most of the spectral energy in natural images is concentrated in the lower end of the spectrum. This should limit the ability of high frequency perturbations to unilaterally change the prediction.

Furthermore, we use linear interpolations of images like in mixup within each band instead of raw images. This closely reflects the more common case where the features in the bands are merely corrupted instead of entirely swapped. It also has the benefit of encouraging linearity inside the same frequency band.

Specifically, the mixing formula for Robustmix is given by

$$\tilde{x} = \texttt{Low}(\texttt{mix}(x_1, x_2, \lambda_L), c) + \texttt{High}(\texttt{mix}(x_1, x_2, \lambda_H), c) \tag{1}$$

$$\tilde{y} = \lambda_c \texttt{mix}(y_1, y_2, \lambda_L) + (1 - \lambda_c)\texttt{mix}(y_1, y_2, \lambda_H) \tag{2}$$

where $\lambda_L, \lambda_H \sim \mathrm{Beta}(\alpha, \alpha)$, α is the Mixup coefficient hyper-parameter, and $\texttt{Low}(\cdot, c), \texttt{High}(\cdot, c)$ are a low pass and high pass filter respectively with a uniformly sampled cutoff frequency $c \in [0, 1]$. And λ_c is the coefficient that determines how much weight is given to the lower frequency band. It is given by the relative amount of energy in the lower frequency band for natural images

$$\lambda_c = \frac{E[\|\texttt{Low}(x_i, c)\|^2]}{E[\|x_i\|^2]}. \tag{3}$$

4 Results

4.1 Datasets and Metrics

The results presented in this paper rely on the mCE measurement on ImageNet-C, the clean accuracies on ImageNet and Stylized-ImageNet (SIN) as well as the shape bias on SIN. These measurements are found in a range of paper studying robustness (Hendrycks & Dietterich, 2018; Hendrycks et al., 2019; Geirhos et al., 2018; Laugros et al., 2020).

ImageNet. We evaluate the common classification accuracy on ImageNet (Deng et al., 2009) which will be referred to as clean accuracy. We use the standard Resnet preprocessing resulting in images of size 224x224 (He et al., 2015). The standard models, without any additional data augmentation process, will be qualified as the baseline.

ImageNet-C. This dataset is made of 15 types of corruption drawn from four main categories: noise, blur, weather and digital (Hendrycks & Dietterich, 2018). These corruptions are applied to the validation images of ImageNet at 5 different intensities or levels of severity. Following (Hendrycks & Dietterich, 2018), we evaluate the robustness of our method by reporting its **mean corruption error (mCE)** normalized with respect to AlexNet errors.

Stylized-ImageNet. Stylized-ImageNet (SIN) is constructed from ImageNet by replacing the texture in the original image using style transfer, such that the texture gives a misleading cue about the image label (Geirhos et al., 2018). The 1000 classes from ImageNet are reduced to 16 shape categories, for instance all labels for dog species are grouped under one dog label, same for chair, car, etc. There are 1280 generated cue conflict images (80 per category). With SIN, we evaluate the classification accuracy (SIN accuracy) and measure the model's shape bias.

4.2 Experimental Setup

We chose to do evaluations on residual nets (ResNet-50 and ResNet-152) and EfficientNets (EfficientNet-B0, EfficientNet-B1, EfficientNet-B5 and EfficientNet-B8). Experiments were run on 8x8 TPUv3 instances for the the bigger EfficientNets (EfficientNet-B5 and EfficientNet-B8); and the other experiments were run on 4x4 TPUv3 slices. For the Resnet models, we use the same standard training setup outlined in Goyal et al. (2017). However, we use cosine learning rate Loshchilov & Hutter (2016) with a single cycle for Resnets that are trained for 600 epochs.

4.3 Robustness Results

Imagenet-C In Table 1, we can see that Robustmix consistently improves robustness to the considered transformations, with a 15 point decrease in mCE over the baseline for ResNet-50. Robustmix gives

the best trade-off between accuracy and robustness over mixup. This can be better observed in Figure 3.

While it is not directly comparable to ViT-L/16 due to its use of $300\times$ more data, we see that Efficientnet-B8 with Robustmix and RandAugment has better robustness at 44.8 mCE. It is also competitive with DeepAugment (Hendrycks et al., 2020) which requires training additional specialized image-to-image models on tasks such as super-resolution to produce augmented images. By comparison, our approach does not rely on extra data or extra trained models.

Method	Clean Accuracy	mCE	Size	Extra Data
ResNet-50 Baseline (600 epochs)	76.3	78.1	26M	0
ResNet-50 BlurPool (Zhang, 2019)	77.0	73.4	26M	0
ResNet-50 Mixup (600 epochs)	**78.2**	67.5	26M	0
ResNet-50 Augmix	77.6	68.4	26M	0
ResNet-50 Augmix + SIN	74.8	64.9	26M	0
ResNet-50 Robustmix (600 epochs)	77.1	**61.2**	26M	0
EfficientNet-B0 Baseline	76.8	72.4	5.3M	0
EfficientNet-B0 Mixup ($\alpha = 0.2$)	**77.1**	68.3	5.3M	0
EfficientNet-B0 Robustmix ($\alpha = 0.2$)	76.8	**61.9**	5.3M	0
BiT m-r101x3 (Kolesnikov et al., 2020)	84.7	58.27	387.9M	12.7M
ResNeXt-101 $32 \times 8d$+DeepAugment+AugMix (Hendrycks et al., 2020)	79.9	**44.5**	88.8M	Extra models
ViT-L/16 (Dosovitskiy et al., 2020)	**85.2**	45.5	304.7M	300M
EfficientNet-B8 Baseline	83.4	60.8	87.4M	0
EfficientNet-B8 Robustmix ($\alpha = 0.4$)	84.4	49.8	87.4M	0
EfficientNet-B8 RandAug+Robustmix ($\alpha = 0.4$)	**85.0**	**44.8**	87.4M	0

Table 1: Comparison of various models based on Imagenet accuracy and Imagenet-C robustness (mCE). The robustness results for BiT and ViT are as reported by Paul & Chen (2021)(Table 3).

Stylized-ImageNet. We confirm that our method indeed increases both accuracy on Stylized ImageNet and the shape bias as shown in table 2. For ResNet-50, Robustmix almost doubles the shape bias from baseline (from 19 to 37) and improves it by 63% over Mixup; while relative improvements on SIN accuracy are of 72% and 33% respectively over baseline and Mixup. The same observation for EfficientNet-B5 wich improves shape bias by near 50% and SIN accuracy by near 60 % over the baseline.

Method/Parameters	SIN Accuracy	Shape Bias
ResNet-50 Baseline	15.6	19.25
ResNet-50 Mixup	20.1	22.7
ResNet-50 Robustmix	**26.8**	**37.0**
EfficientNet-B5 Baseline	25.3	44.4
EfficientNet-B5 Mixup	28.75	48.3
EfficientNet-B5 Robustmix	**40.3**	**66.1**

Table 2: Accuracy and shape bias computed on Stylized Imagenet.

5 Conclusion

Our method offers a promising new research direction for robustness with a number of open challenges. We have used a standard DCT based low-pass filter on images and L2 energy metric to determine the contribution of each label. This leaves many alternatives to be explored, such as: different data modalities like audio; more advanced frequency separation techniques like Wavelets; and alternative contribution metrics for mixing labels.

References

References

Athalye, A., Engstrom, L., Ilyas, A., and Kwok, K. Synthesizing robust adversarial examples. In *International conference on machine learning*, pp. 284–293. PMLR, 2018.

Beckham, C., Honari, S., Verma, V., Lamb, A. M., Ghadiri, F., Hjelm, R. D., Bengio, Y., and Pal, C. On adversarial mixup resynthesis. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/f708f064faaf32a43e4d3c784e6af9ea-Paper.pdf`.

Cheng, T.-Y., Yang, H.-R., Trigoni, N., Chen, H.-T., and Liu, T.-L. Pose adaptive dual mixup for few-shot single-view 3d reconstruction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 427–435, 2022.

Chuang, C.-Y. and Mroueh, Y. Fair mixup: Fairness via interpolation. *arXiv preprint arXiv:2103.06503*, 2021.

Deng, J., Dong, W., Socher, R., Li, L., Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

Faramarzi, M., Amini, M., Badrinaaraayanan, A., Verma, V., and Chandar, S. Patchup: A regularization technique for convolutional neural networks. *arXiv preprint arXiv:2006.07794*, 2020.

Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F. A., and Brendel, W. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2018.

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

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Guo, C., Frank, J. S., and Weinberger, K. Q. Low frequency adversarial perturbation, 2018. URL `https://arxiv.org/abs/1809.08758`.

Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., et al. Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*, 2014.

Hasanpour, S. H., Rouhani, M., Fayyaz, M., and Sabokrou, M. Lets keep it simple, using simple architectures to outperform deeper and more complex architectures. *arXiv preprint arXiv:1608.06037*, 2016.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition, 2015.

Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

Hendrycks, D., Mu, N., Cubuk, E. D., Zoph, B., Gilmer, J., and Lakshminarayanan, B. Augmix: A simple data processing method to improve robustness and uncertainty. In *International Conference on Learning Representations*, 2019.

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

Kim, J.-H., Choo, W., and Song, H. O. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pp. 5275–5285. PMLR, 2020.

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., and Houlsby, N. Big transfer (bit): General visual representation learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V 16*, pp. 491–507. Springer, 2020.

Laugros, A., Caplier, A., and Ospici, M. Addressing neural network robustness with mixup and targeted labeling adversarial training, 2020.

Li, X.-C., Zhang, X.-Y., Yin, F., and Liu, C.-L. F-mixup: Attack cnns from fourier perspective. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pp. 541–548. IEEE, 2021.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

Mai, Z., Hu, G., Chen, D., Shen, F., and Shen, H. T. Metamixup: Learning adaptive interpolation policy of mixup with metalearning. *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

Mao, X., Qi, G., Chen, Y., Li, X., Duan, R., Ye, S., He, Y., and Xue, H. Towards robust vision transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12042–12051, 2022.

Moosavi-Dezfooli, S.-M., Fawzi, A., Fawzi, O., and Frossard, P. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1765–1773, 2017.

Paul, S. and Chen, P.-Y. Vision transformers are robust learners. *arXiv preprint arXiv:2105.07581*, 2021.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

Verma, V., Lamb, A., Beckham, C., Najafi, A., Mitliagkas, I., Lopez-Paz, D., and Bengio, Y. Manifold mixup: Better representations by interpolating hidden states. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 6438–6447. PMLR, 09–15 Jun 2019. URL `https://proceedings.mlr.press/v97/verma19a.html`.

Wang, H., He, Z., Lipton, Z. C., and Xing, E. P. Learning robust representations by projecting superficial statistics out, 2019. URL `https://arxiv.org/abs/1903.06256`.

Wang, H., Wu, X., Huang, Z., and Xing, E. P. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8684–8694, 2020.

Yang, Y. and Soatto, S. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4085–4095, 2020.

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

Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., and Yoo, Y. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.

Zhang, R. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pp. 7324–7334. PMLR, 2019.

A Appendix

A.1 Ablation Study

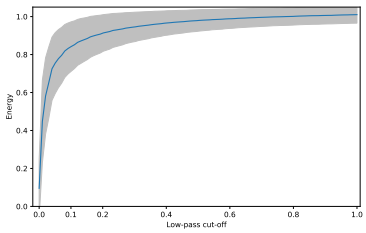
In order to measure the effect of Robustmix we apply some simplifications both with respect to the image mixing as well as the labelling. The results are compiled in table 3. It can be noticed from the first two lines that ablating the energy weighting results in a significant drop in mCE even though there is a slight accuracy improvement. However, keeping the energy weighting but not applying the inband mixups is largely detrimental both to accuracy and robustness. These results show that Robustmix achieves a better combination of mCE and accuracy than these ablations.

Method	Mixed Image	Label	Test Accuracy	mCE
Robustmix - Full (inband mixups and energy weighting)	Equation 1	Equation 2	77.1	61.2
Robustmix without energy weighting	Equation 1	Equation 2 with λ_c replaced by c	77.6	67.7
Robustmix without inband mixups and with energy weighting ($\lambda_L = 1, \lambda_H = 0$)	$\texttt{Low}(x_1, c) + \texttt{High}(x_2, c)$	$\lambda_c y_1 + (1 - \lambda_c) y_2$	68.6	75.3
Robustmix without inband mixups and without energy weighing ($\lambda_L = 1, \lambda_H = 0$ and cutoff c as label coefficient)	$\texttt{Low}(x_1, c) + \texttt{High}(x_2, c)$	$c y_1 + (1 - c) y_2$	74.8	77.5

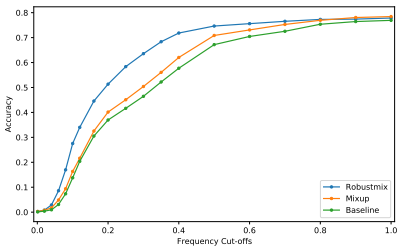
Table 3: Comparison of Robustmix with simplified cases. The results are reported on ResNet50.

A.2 Analysis and Discussion

Low frequency bias In order to quantify the degree to which models rely on lower frequencies, we measure how much accuracy drops as we remove higher frequency information with a low-pass filter. Figure 2b shows that Robustmix is comparatively more robust to the removal of high frequencies. This indicates that models trained with Robustmix rely significantly less on these high-frequency features to make accurate predictions.



(a) Plot of the cumulative energy in Imagenet images as a function of the frequency cutoff.



(b) Test accuracy on Imagenet samples passed through a low-pass filter with increasing cut-off.

Figure 2: As expected, we observe on fibure (b) that Robustmix is more robust to the removal of high frequencies than Mixup. The comparison is done here on ResNet-50 models.

A.3 Extras

Implementation The frequency separation is implemented using a Discrete Cosine Transform (DCT) to avoid the complex multiplication required by a Discrete Fourier Transform. Additionally, we must apply the DCT transform over the x and y dimension separately. Thus, 6 DCT matrix multiplications are required which results in 0.2 GFLOPs per image. In contrast, just the forward pass of ResNet50 requires 3.87 GFLOPs (Hasanpour et al., 2016).

In our implementation of Robustmix, we reorder commutative operations (low pass and mixing) in order to compute the DCT only a single time per minibatch. The pseudocode is provided in Algorithm 1, where reverse is a function that reverses the rows of its input matrix.

Algorithm 1 Robustmix

Input: Minibatch of inputs $X \in \mathbb{R}^{N \times H \times W \times D}$ and labels $Y \in \mathbb{R}^{N \times C}$, $\alpha \in \mathbb{R}$
Output: Augmented minibatch of inputs $\tilde{X} \in \mathbb{R}^{N \times W \times H \times D}$ and labels $\tilde{Y} \in \mathbb{R}^{N \times C}$
$\lambda_L, \lambda_H \sim \text{Beta}(\alpha, \alpha)$ and $c \sim U(0, 1)$
$L \leftarrow \texttt{Low}(X, c)$
$H \leftarrow 1 - L$
$\lambda_c \leftarrow \frac{\|L\|^2}{\|X\|^2}$
$\tilde{X} \leftarrow \texttt{mix}(L, \text{reverse}(L), \lambda_L) + \texttt{mix}(H, \text{reverse}(H), \lambda_H)$
$\tilde{Y} \leftarrow \texttt{mix}(Y, \text{reverse}(Y), \lambda_c * \lambda_L + (1 - \lambda_c) * \lambda_H)$

Choice of hyperparameter α

In our cross-validation of α, we found small values less than 0.2 perform poorly both on accuracy and mCE. Values of α such that $0.2 \leq \alpha \leq 0.5$ not only give the best accuracies and mCEs but also the best trade-off of mCE versus accuracy as bigger values of α have shown giving good values for accuracy but do not do as well on mCE. In our experiments, we found that we typically achieve good results with a frequency cutoff c sampled between $[0, 1]$ as described in Algorithm 1. However, For ResNet-50 trained with a training budget that is too limited (200 instead of 600 epochs) and its smaller versions (ResNet-18 and ResNet-34), it can be beneficial to fix a minimum $c \geq \tau$ for the cutoff by sampling in the interval $[\tau, 1]$. The minimum cutoff determines the range at which band mixing will occur.

Method	Clean Accuracy	mCE	Size	Extra Data
ResNet-50 Baseline (200 epochs)	76.3	76.9	26M	0
ResNet-50 Baseline (600 epochs)	76.3	78.1	26M	0
ResNet-50 BlurPool (Zhang, 2019)	77.0	73.4	26M	0
ResNet-50 Mixup (200 epochs)	77.5	68.1	26M	0
ResNet-50 Mixup (600 epochs)	**78.2**	67.5	26M	0
ResNet-50 Augmix	77.6	68.4	26M	0
ResNet-50 Augmix + SIN	74.8	64.9	26M	0
ResNet-50 Robustmix (600 epochs)	77.1	**61.2**	26M	0
EfficientNet-B0 Baseline	76.8	72.4	5.3M	0
EfficientNet-B0 Mixup ($\alpha = 0.2$)	**77.1**	68.3	5.3M	0
EfficientNet-B0 Robustmix ($\alpha = 0.2$)	76.8	**61.9**	5.3M	0
EfficientNet-B1 Baseline	78.1	69.4	7.8M	0
EfficientNet-B1 Mixup ($\alpha = 0.2$)	**78.9**	64.7	7.8M	0
EfficientNet-B1 Robustmix ($\alpha = 0.2$)	78.7	**57.8**	7.8M	0
EfficientNet-B5 Baseline	82.7	65.6	30M	0
EfficientNet-B5 Mixup ($\alpha = 0.2$)	83.3	58.9	30M	0
EfficientNet-B5 Robustmix ($\alpha = 0.2$)	83.3	51.7	30M	0
EfficientNet-B5 RandAug+Robustmix ($\alpha = 0.2$)	**83.8**	**48.7**	30M	0
BiT m-r101x3 (Kolesnikov et al., 2020)	84.7	58.27	387.9M	12.7M
ResNeXt-101 $32 \times 8d$+DeepAugment+AugMix (Hendrycks et al., 2020)	79.9	**44.5**	88.8M	Extra models
ViT-L/16 (Dosovitskiy et al., 2020)	**85.2**	45.5	304.7M	300M
RVT-B^* (Mao et al., 2022)	**82.7**	46.8	91.8M	PAAS+Patch-wise[1]
EfficientNet-B8 Baseline	83.4	60.8	87.4M	0
EfficientNet-B8 Robustmix ($\alpha = 0.4$)	84.4	49.8	87.4M	0
EfficientNet-B8 RandAug+Robustmix ($\alpha = 0.4$)	**85.0**	**44.8**	87.4M	0

Table 4: Comparison of various models based on Imagenet accuracy and Imagenet-C robustness (mCE). The robustness results for BiT and ViT are as reported by Paul & Chen (2021)(Table 3).

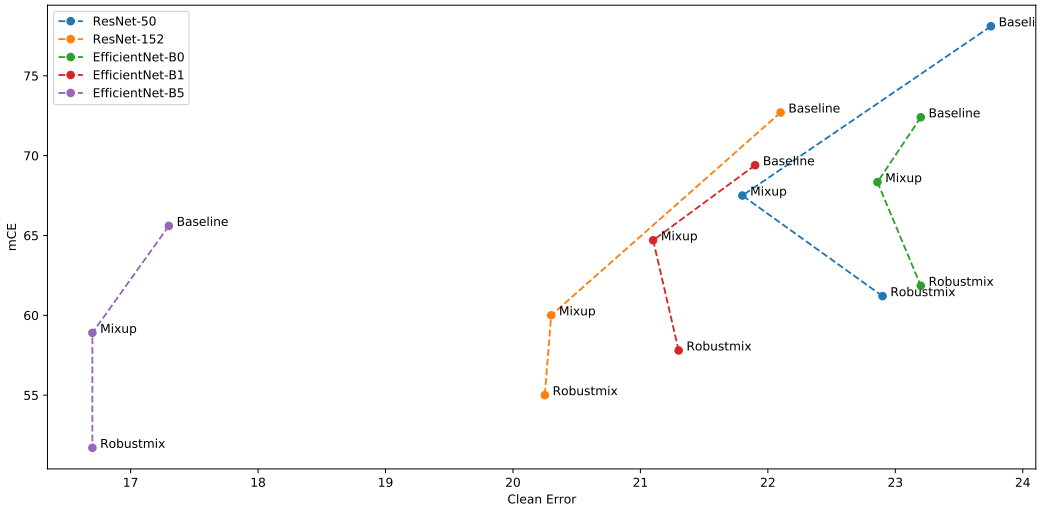


Figure 3: Highlighting the tradeoff between mCE and Clean Error for various models.

A.4 Robustness to Adversarial Perturbations

In this section, we consider robustness to adversarial perturbations (Goodfellow et al., 2014). Adversarial perturbations make no visually distinguishable difference to the human eye but make models misclassify examples. Wang et al. (2020) and Yin et al. (2019) have shown that adversarial perturbations of unregularized models disproportionally affect higher frequencies. As we have seen in Figure 2b, Robustmix encourages the model to rely more on the lower frequencies to make predictions. Our hypothesis is that this will have a beneficial effect on adversarial robustness. For our experiment we use one of the first methods proposed in the Deep Learning community to construct adversarial examples, the *"Fast Gradient Sign Method"*(FGSM) (Goodfellow et al., 2014). The adversarial example is constructed by adding a perturbation proportional to the sign of the gradient of the loss with respect to the input: $\mathbf{x} = \mathbf{x} + \epsilon \operatorname{sign}\left(\Delta_{\mathbf{x}} J(\theta, \mathbf{x}, y)\right)$. As seen in Figure 4, we find Robustmix is more robust than the baselines to this type of adversarial attack.

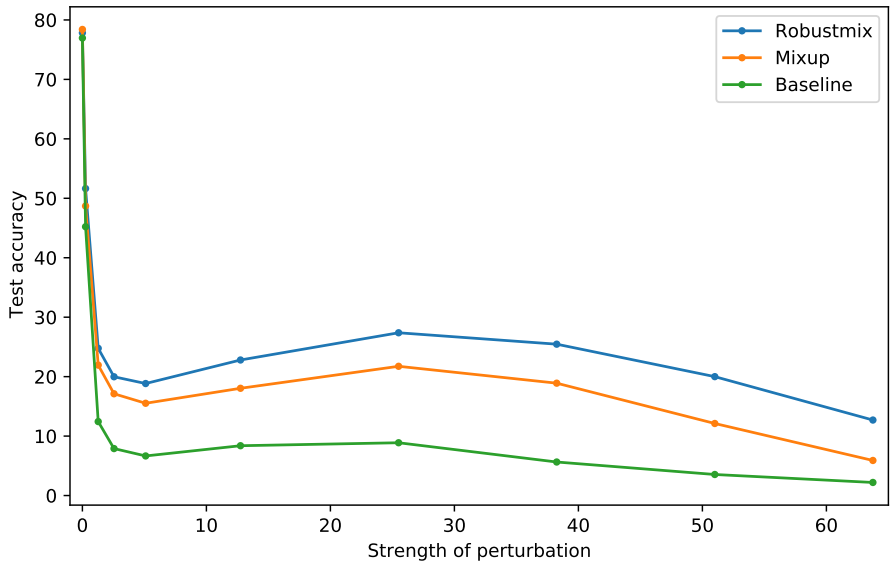


Figure 4: Test accuracy w.r.t. increasing adversarial perturbation strength. The comparison is done here on ResNet-50 models for baseline, Mixup and Robustmix.