Learning the *Essential* in less than 2k additional weights - a single large kernel convolution layer can improve prediction stability under unknown corruptions

Anonymous CVPR submission

Paper ID 16292

Abstract

The performance of image classification on well-known benchmarks such as ImageNet is remarkable, but in safety-critical situations, the accuracy often drops significantly under adverse conditions. To counteract these performance drops, we propose a very simple modification to the models: we pre-pend a single, dimension preserving convolutional layer with a large linear kernel whose purpose it is to extract the information that is essential for image classification. We show that our simple modification can increase the robustness against common corruptions significantly, especially for corruptions of high severity. We demonstrate the impact of our channel-specific layers on ImageNet-100 and ImageNette classification tasks and show an increase of up to 30% accuracy on corrupted data in the top1 accuracy. Further, we conduct a set of designed experiments to qualify the conditions for our findings. Our main result is that a data- and network-dependent linear subspace carries the most important classification information, which our proposed pre-processing layer approximately identifies for most corruptions, and at very low cost.

1. Introduction

Intensive research into DNN architectures [19, 36, 51, 53], improved for example by Neural Architecture Search (NAS) [9, 53] and advanced training schemes [6, 57], has produced impressive classification results [9, 12]. The performance of models on well-known benchmarks such as ImageNet [46], Cifar-100 [30] and others has improved significantly over the last decade. However, a persistent challenge arises when these systems are exposed to adverse conditions, such as changes in lighting, weather and other optical corruptions [20, 39]. Despite achieving high accuracy on in-domain data, DNNs often experience a significant drop in performance under these real-world challenges [21, 39, 48]. Intensive research is therefore being carried out into methods to increase robustness to various disturbances.

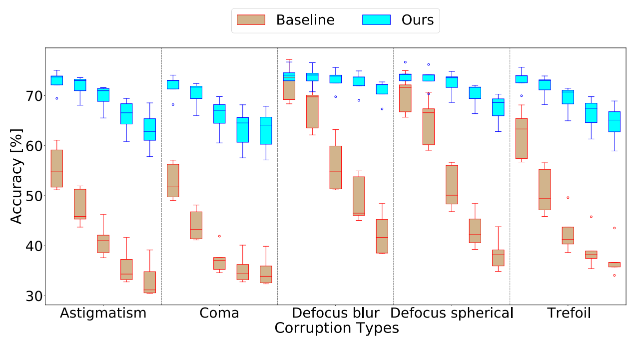


Figure 1. Convolutional preprocessing (ours) can increase the robustness of classification networks against unknown corruptions *without data augmentation*. ResNet50 improved with a trainable preprocessing filter evaluated on ImageNette [27] blur and corruptions from OpticsBench [39]. For each corruption type, five levels of severity are shown from left to right. The variation, visualized via the box plots, results from five different seeds per model.

A common strategy to increase robustness is to appropriately augment the original training dataset with relevant diversification. Such data augmentation techniques include geometric transformations, cutouts and mixing of images [66], color jitter or the simulation of out-of-distribution data by introducing common corruptions [20], and optical corruptions [39]. Other methods involve adversarial training, whose objective is yet at odds with robustness to some real-world corruption types [63].

In this paper, we propose a very simple yet effective trick to improve model generalization, which consists of pre-pending to the model a single large kernel depth-wise convolution operation without strides. The proposed layer is trained with the model and can, in principle, learn a complete representation of the image (*e.g.* with an identity mapping), but no over-complete one. This is in contrast to the usual first model layers that create over-complete representations to facilitate sparse coding. Surprisingly, we find that our simple input layer trained solely on clean data without particular augmentation strategies increases classification robustness to various corruption types on multiple DNN architectures by up to 9.8% across 21 corruptions and by over 30% on specific severities with less than 2k additional parameters (*e.g*. only 0.008% for ResNet50). See Fig. 1 for an example. We thoroughly investigate this outcome with different methods to learn such large per-color-channel, *i.e*. dimension preserving, kernels, and compare to the respective baseline trained without the extra layer.

Major Findings and Contributions Our empirical study indicates that the proposed, dimension-preserving large kernel input layer, while being able to learn a complete data representation, tends to learn a subspace projection. As such, it extracts the crucial content from the input training samples, *i.e*. the *essential*, such as to preserve a high accuracy on clean data. The dropping of non-essential parts of the signal, *i.e*. the learned subspace projection, automatically leads to an increase in the model's generalization ability. We show this on ImageNette [27] and ImageNet-100 [55] for the corruption types from [20] and [39] across diverse image classification models. Further, we explore in a signal theory-inspired study the properties of suitable kernels for the proposed layer, such as to gain deeper understanding and foster future progress in this very cost efficient direction of improving model generalization.

2. Related Work

Model robustness and stability have been discussed under various perspectives. In the following, we first give a brief overview on standard benchmarks for the evaluation of classification robustness under corruptions, then, we summarize related work on model hardening through adversarial training. The proposed method differs significantly from these approaches, as no assumptions of potential threats to the model are made during training. Instead, it implicitly encourages the model to learn the relevant signal content while reducing parts of the input data that are less relevant (*i.e*. noise). To contextualize this finding, we also discuss prior work on the interplay between learned frequencies and model robustness, as well as prior art on image resampling for neural networks.

Image Corruptions and Data Augmentations To improve classification robustness against corruptions, data augmentation can be used to mimic the diversification of real data. AugMix [21] improves robustness to common corruptions, [39] use optical blur kernels to additionally improve accuracy on primary optical aberrations. Others add more abstract augmentations with image combinations [2, 3, 66], feature map perturbations [22], perturbed frequency representations [65], or by adapting/augmenting the style of training images [25, 33, 62, 69] using SotA generative models [24]. All these methods significantly increase the training time of a model. The robustness they provide is limited to corruptions that are similar to the augmented data. In comparison, our models are trained only on the clean dataset, avoiding i) the guessing of the corruptions, and ii) the computational overhead of an augmented data set, while offering improved generalization ability of the trained model in many settings.

Adversarial Attacks and Training Corruption benchmarks, *e.g*. [20, 48], allow testing the model behavior w.r.t. predetermined corruption types. In contrast, adversarial attacks can add any (usually ϵ-bounded) perturbation. They usually assume Lipschitz continuity of robust models [1, 4, 15, 28, 31, 60]. When used during training, adversarial samples can be employed to harden a model [11, 15, 44, 59, 61, 68, 68], where some strategies involve additional loss terms [11, 68] or training data [5, 16, 49] (*e.g*. 1M extra samples generated by [24] are used in [16, 41, 43] for adversarial training on CIFAR-10). While significantly improving model robustness to adversarial samples, the additional training costs are immense. The required compute resources increase *e.g*. by a factor of five to 15 even if the simple strategy of using adversarial samples during training is employed. Further, it has been discussed *e.g*. in [13, 47, 63] that adversarial training is at odds with model generalization to some real-world corruption types, which we focus on in this work. In contrast to the above discussed methods, our approach only requires a negligible overhead of less than 2k additional parameters and our models are trained using the respective standard training parameters, *i.e*. there are negligible extra-costs, while improving model robustness to various common corruptions.

Learned Frequencies and Robustness Previous works have studied the effect of learned frequencies within the shallow and deep layers of neural networks on model robustness (*e.g*. [63]). In [47], it is shown that regularizing a model to learn low frequencies and high frequencies separately can improve robustness to common corruptions. [18] demonstrated a correlation between aliasing in CNN downsampling layers and their susceptibility to adversarial attacks. Several approaches reduce or remove aliasing in the downsampling operators to improve the learned representations and their robustness [17, 26, 32, 70, 72]. Further, [14] showed that CNNs tend to focus on image textures rather than shapes to determine an object class and [13] discuss how adversarial training can shift this bias towards shapes with both positive and negative effects on model robustness to common corruptions, depending on the corruption type. In contrast to these works, our approach solely focuses on the first model layer and no specific bias towards high or low frequencies is added. Yet, by providing only a single large kernel where the number of output channels equals the number of inputs, we implicitly encourage the model to focus on the essential part of the data.

Large Kernel CNNs in the "Era of VITs" Vision Transformers [10] (ViT) trained on huge amounts of data have recently been outperforming classical small kernel CNNs [19] on standard benchmarks, causing the community to shift its focus towards further improving transformer based architectures [7, 64, 67]. Contrarily, this has led to a trend towards increasing the filter size within deep layers of CNNs, *e.g.*, it was shown that even kernels with 7×7 convolutions in CNNs can allow them to outperform [36] self-attention based vision transformers [10, 35]. Extending on [56], Smith et al. [50] very recently show that CNNs perform on par with vision transformers at scale. Our approach is therefore evaluated on CNNs, because they reach high accuracies even when trained on rather low amounts of data, which facilitates our in-depth study.

In [8, 34, 40], the concept of large kernel CNNs is expanded with kernels up to sizes of 51×51 within the network, where handling the memory consumption is a challenge. Global filter networks [42] apply the filter in the frequency domain to allow for infinitely extended filters. These cases further highlight the benefit of using large kernels within the network. While we are also using a large kernel, our approach is different from the above: We use this filter only for the model input and perform an image-to-image mapping with it, encouraging the model to summarize the important information in the data, so that the entire model remains light-weight and can be trained with a low amount of data - yet improves robustness.

Learning CNN Inputs Several works have proposed to resample data in a non-uniform way at the model input or in deep layers to allow for precise predictions in regions of interest, *e.g.* [23, 29, 54, 71]. Such approaches aim for individual downsampling patterns for every image. In contrast, our approach treats all images equally and operates under the stationarity assumption, *i.e.* the applied convolution filter is constant over the entire image. Since downsampling can lead to aliasing, other works propose to learn to uniformly downsample so that more information is preserved [38, 52, 58]. They all aim to improve the model's prediction accuracy. In particular, [52] propose to optimize a small deep neural network for the downsampling task, where the output of the network is restricted to be an image. Similarly, the output of our single layer is an image. Yet, our layer does not perform any resampling, avoiding potential aliasing, nor does it provide any non-linearity and can therefore be analyzed using linear techniques.

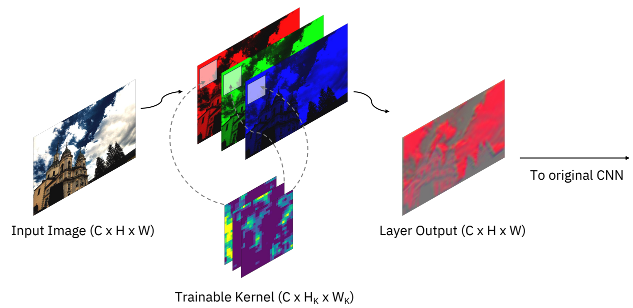


Figure 2. The architecture of our proposed trainable input layer. We learn a single depth-wise convolution to encourage the model to represent the essential.

3. Enhancing Prediction Stability with a Trainable Convolution Input Layer

Our aim is to investigate a simple approach to improve a model's stability under corruptions without increasing the training load. Our approach follows the intuition that it is beneficial to encourage a model to learn the relevant information from the data while neglecting superfluous parts of it, *e.g.* noise. To do so, we propose to add an extra convolution layer with a large kernel in front of the input layer of the model, where the output dimensions equal the input dimensions, *i.e.* no overcomplete input representation can be learned. If needed, the model can thus learn to preserve all data. Since, however, not all parts of the training data are valuable for the classification process (*e.g.* noise), we hypothesize that less important parts will likely be dropped in the learned mapping. The architecture, shown in Fig. 2, uses one kernel for each input channel. The input to the layer is the to-be-classified image with C image color channels and a size of $H \times W$. The kernel size of the convolutional input layer is $C \times H_K \times W_K$ with a stride of 1, *i.e.* the result of the depth-wise convolution has the same dimensions, as the input image. In contrast to a typical convolutional layer, this input layer does not fuse the information of the color channels. The layer's output is propagated to the first layer of the otherwise unchanged DNN model, without additional non-linearity applied. Empirically, as shown for example in Fig. 1, our simple approach leads to remarkable results.

Intuitively, the extra layer performs a specific linear transformation that can shift and/or block or emphasize the color-dependent content of an image. The data range of the kernel is not limited to positive numbers, so negative kernel values can also *sharpen* image content. This raises the question, which parts of the input data are preserved in our layer, *i.e.* whether the layer acts, for example, as an amplifier by spatially distributing the information in a better way, or whether it acts as a projection layer, where particular parts of the data are explicitly dropped. In the following, we propose a systematic approach to empirically test these options, by considering three different kernel classes.

Study Design The two key characteristics of a linear transformation are a) its rank and b) its conditioning, *i.e.* its noise amplification characteristics in the case of a nominally full-rank transformation. The latter is characterized by the condition number (CN) of the transformation. Let $\cdot * g$ be the linear transformation effected by a convolution with the kernel g, and $\cdot * g^{-1}$ its inverse, then

$$CN(\cdot * g) = \frac{\lambda_{\text{max}}}{\lambda_{\text{min}}} = \frac{1/\lambda_{\text{min}}}{1/\lambda_{\text{max}}} = CN(\cdot * g^{-1}),$$

where λ_{min} and λ_{max} are the minimum and the maximum eigenvalues of the transformation. The equation shows that the forward kernel and the inverse kernel have the same condition number, which explains our use of the term *noise amplification*. We use the condition number as a numerical indicator of the preservation of signal content[1].

While the linear behavior of our proposed pre-processing layer is well understood, the reaction of the subsequent non-linear network architecture to this modified input, is not. Our intention is to study the effect of the above-mentioned kernel properties on classification robustness.

We therefore introduce three kernel classes for further studying the properties of the proposed convolutional pre-processing layer, with the underlying hypothesis that signal content preservation or removal is a decisive factor in the observed robustness increase.

name	CN	rank
class I (content preserving)	unity/low	full
class II (fully trained/static)	medium	full
class III (projection-type)	large/infinite	rank deficient

Table 1. Categories of kernels with their corresponding condition number (CN) and rank.

Class I: Content Preserving Kernels A minimal noise amplifying kernel is one whose associated linear transformation has a determinant of one: such transformation is unitary, *i.e.* vector norms are not changed.

For a convolution kernel, the associated matrix is a circulant matrix with the convolution kernel in the rows (assuming circular boundary conditions). The eigenvectors of circulant matrices form the Fourier basis, which is the underlying reason for the convolution theorem. The associated eigenvalues are the Fourier coefficients. Since the product of the eigenvalues yields the determinant of the linear transformation, we see that all Fourier coefficients must have unit amplitude for the determinant to be unity. Since the Fourier coefficients are complex, they can still have arbitrary phases while fulfilling the unit amplitude constraint. An additional consideration, however, is that the associated kernels be real. This forces the constraint that $G(-\omega) = G^*(\omega)$ with $G(\omega)$ being the Fourier spectrum, parameterized by angular frequency component ω, of the spatial kernel function $g(x)$. The constraint implies that we have $N/2$ real degrees of freedom to construct N pixel content-preserving kernels, where $N = H_K \times W_K$. We use this parameterization for our practical layer implementation. A unit condition number can only be achieved for kernels of the same size as the image (225×225). We also experiment with smaller kernels (25×25) of the same construction, also referring to them as content preserving even though the condition number of the associated linear transformation is a low number above unity ($10^2 - 10^3$).

Class II: Fully Trained Kernels are parameterized by their real value entries in the spatial domain. Positive and negative values are allowed, but not complex ones. In the absence of the aforementioned constraints of *class I* kernels, we are free to choose the size of the kernels. In order to consistently evaluate the effects of the remaining kernel types, we choose this to be 25×25. We have observed that such freely trained kernels yield condition numbers in an intermediate range of $10^4 - 10^5$. For comparison, we also include a number of static kernels in *class II*.

Class III: Projection-type Kernels have a CN that is (numerically) infinite, since at least one Fourier coefficient is (numerically) zero. The removed subspace dimension is equivalent to the number of zero Fourier coefficients in the kernel's spectrum. We explore two ways to generate such kernels.

First, we explore whether low-value Fourier coefficients in the fully learned kernels of *class II* can be replaced by zeros (thresholding), implying that the low values found by the optimization algorithm are effectively sufficient to remove the subspace in question from the data for all purposes of the nonlinear network part.

Second, we encourage zero Fourier coefficients by means of an additional $L1$-regularization on the Fourier coefficients of the fully trained kernel. The associated sparsity then encourages projection-type kernels.

The interesting characterizing number for projection-type kernels is the dimensionality of the null-space of the kernel, *i.e.* the number of its zero Fourier coefficients. A larger number indicates a higher rate of signal content removal.

4. Experimental Evaluation

In the following, we evaluate the DNN prediction stability with our proposed, trainable image-to-image convolution input layer. The DNNs we evaluate are trained on different publicly available subsets of ImageNet [45] to allow for extensive experiments. ImageNette [27] is a dataset consisting of 10 ImageNet classes. It has 9,469 training and 3,925 validation images [27]. ImageNet-100 [55] uses 100 ImageNet classes with a total of 128k training and 5,000 validation images [55].

All baseline models and all models with additional convolution input layer are trained on clean data, *without* any data augmentation. To ensure a comparability between the baseline and our trained models, we only add our

[1] We emphasize that we are not arguing in an information-theoretic, but in a numerical sense.

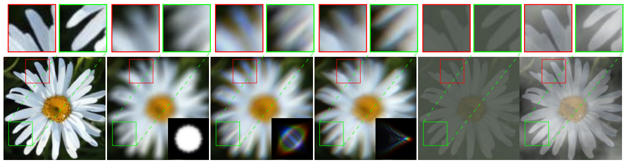


Figure 3. Overview of different corruptions from OpticsBench's primary aberrations (astigmatism, coma) [39] and rotationally symmetric defocus blur, contrast and fog from [20] applied to an ImageNet sample. OpticsBench's astigmatism and coma introduce chromatic aberration (visible at the flower petals) and directional blur, which can challenge DNNs differently than rotational symmetric luminance blur. All corruptions are shown in the supplementary material Sec. 7.

Model	Version	CD	OB [39]	CC [20]
ResNet50	Base	***0.800**	*0.592	*0.487
ResNet50	Trainable	*0.775	***0.685**	***0.565**
AlexNet	Base	***0.848**	*0.572	*0.605
AlexNet	Trainable	*0.838	***0.670**	***0.660**
EfficientNet	Base	**0.907**	0.604	0.605
EfficientNet	Trainable	0.903	**0.629**	**0.633**
MobileNet	Base	**0.897**	0.589	0.564
MobileNet	Trainable	0.893	**0.639**	**0.611**
DenseNet161	Base	**0.898**	0.547	0.535
DenseNet161	Trainable	0.885	**0.605**	**0.597**

Table 2. Results on ImageNette for conventionally trained DNNs (baseline) and additional fully trainable layer. CD= Clean Data, OB = OpticsBench, CC = Common corruptions. * = average from multiple seeds. The results on the corruption benchmarks are averaged across severity and corruption. The highest accuracy per DNN is marked in bold.

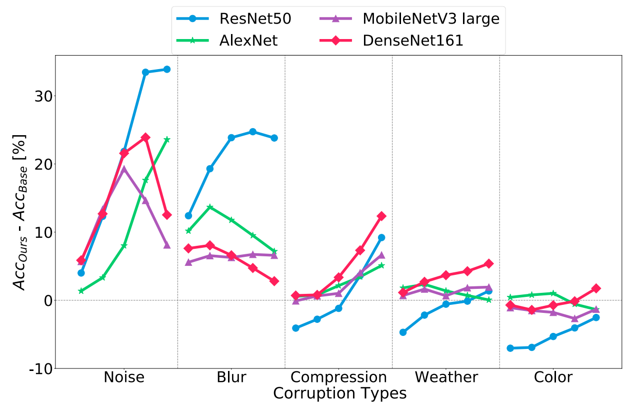


Figure 4. Relative accuracy improvements on ImageNette [27]. A fully trainable input layer (class II) can increase the robustness of classification networks against unknown corruptions *without data augmentation*. We evaluate different DNNs on blur and noise corruptions from OpticsBench [39] and 2D Common Corruptions [20]. For each corruption type, five levels of severity are shown from left to right.

proposed input layer to the corresponding model and do not change any hyperparameters. The full details of chosen hyperparameters are given in the supplementary material Sec. 8. The implementation is based on PyTorch and the training recipes follow [37]. In order to draw a more comprehensive picture, some experiments were trained on five different seeds. Subsequently, all models are evaluated on clean data and on two corrupted datasets [20, 39]. The performance of these trainings can be examined in Table 2, 3, and 4. Furthermore, we trained our proposed model in an adversarial fashion and evaluate it against the baseline in the supplementary material Sec. 9.6.

To test against corruptions, we use the common corruptions from Hendrycks et al. [20], which are each binned into five different severities and apply them to ImageNette and ImageNet-100. To test for more diverse blur types, we also include the OpticsBench from Müller et al. [39], which covers primary optical aberrations and is similarly organized. The blur kernels are size-matched to the defocus-blur-corruption kernels from [20]. Fig. 3 gives a visual impression of the corruptions that we evaluate.

For the sake of readability, we summarize all different corruptions into five super-categories (noise, blur, compression, weather, color) and take the average of all the subcategories. We provide the full details and figures with all individual corruptions in the supplementary material Sec. 9.

4.1. Trainable Large Kernels can Improve Prediction Stability

In the following, we compare the prediction stability under corruption of models with our proposed trainable convolutional pre-processing filter to the respective baselines. First, the *class II* kernels are evaluated on the ImageNette corruptions. In Fig. 4, we plot the performance relative to the baseline for better readability across different model families. Positive values indicate an improvement over the baseline model, and negative values indicate worse predictions. Absolute accuracies averaged over OpticsBench and Common Corruptions and on clean data are given in Table 2.

Looking at all categories of corruption in Fig. 4, it is noticeable that blur and noise benefit significantly from the additional input filter compared to the baseline. For the other types, the accuracies are on par with the baseline, and in some cases even get slightly below. This is especially true for color corruptions, which is to be expected since our proposed depth-wise convolutions can not learn color recombinations. Overall, yet in particular for ResNet50, the relative performance of the trainable input layer improves with increasing severity for both noise and blur. As seen in Table 2, the prediction accuracy improves by a large margin on OpticsBench [39] and Common Corruptions [20] for all models, *e.g.* 9% on OpticsBench and 8% on common cor-

Model	Version	CD	OB [39]	CC [20]
ResNet50	Base	***0,801**	*0.536	*0.406
ResNet50	Trainable	*0.797	***0.558**	***0.437**
AlexNet	Base	**0.698**	0.339	0.307
AlexNet	Trainable	0.671	**0.363**	**0.344**
EfficientNet	Base	**0.796**	0.480	**0.395**
EfficientNet	Trainable	0.795	**0.509**	0.393
MobileNet	Base	**0.780**	0.470	0.344
MobileNet	Trainable	0.761	**0.501**	**0.404**

Table 3. Results on ImageNet-100 for conventionally trained DNNs (baseline) and additional fully trainable layer. CD= Clean Data, OB = OpticsBench, CC = Common corruptions. * = multiple seeds. The results on the two corruption benchmarks are averaged across severity and corruption.

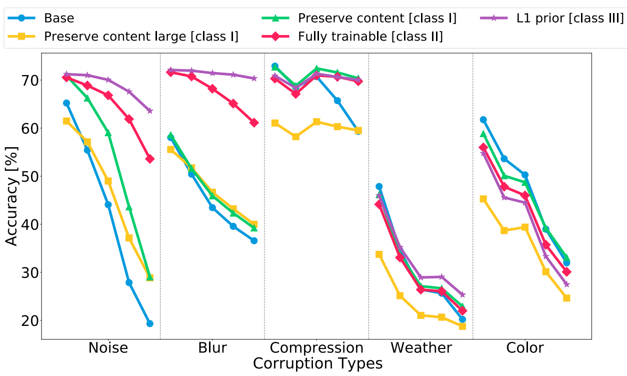


Figure 5. Comparison of different kernel types on corrupted ImageNette data for ResNet50. For additional kernels, see Table 4.

ruptions for ResNet50, while the accuracy on clean data is only slightly decreased.

Second, the same experiment is performed on ImageNet-100 with the same types of corruption. Table 3 lists the achieved accuracies on clean data, OpticsBench [39] and common corruptions [20]. The trainable layer improves again on average over the corruptions for each DNN. However, compared to the results on ImageNette, the improvements are now smaller. ResNet50 with the trainable large input kernel performs best with an increase in accuracy of $+2.2\%$ on OpticsBench and $+3.1\%$ on common corruptions. More results on ImageNet-100 are given in the supplementary material Sec. 9.5. In the following, we investigate with help of our proposed kernel *classes I* and *III* which properties the learned kernels have. Understanding these properties is particularly intriguing because they allow the kernels to represent the input data without an increase in dimension, while the subsequent networks generalize better to many unseen corruption types.

4.2. Which properties of trainable kernels can help? – Comparing Kernel Classes

The results in Fig. 4, Table 2 and 3 show an improvement in prediction over many corruption types. This raises the question of what kernel properties cause these results, and whether they can be improved further with different kernels. In order to deepen the considerations from Sec. 3, we analyze numerous variants of the input layer kernel for a ResNet50 and compare the different kernel *classes I-III*. The results for more models are given in the supplementary material Sec. 9. Table 5 lists the absolute accuracies.

Kernel type	class	CD	OB [39]	CC [20]
None (Base)	-	***0.800**	*0.592	*0.487
Preserve content	I	0.754	0.476	0.513
Preserve content large	I	0.645	0.478	0.440
Conv2D KS=25	II	0.655	0.601	0.501
Fully trainable	II	*0.775	*0.685	*0.565
Random initialization	II	0.711	0.673	0.550
L1 prior	III	0.712	**0.699**	**0.567**
Directional blur filter	II	0.778	0.437	0.345
Gauss blur filter	II	0.764	0.668	0.541

Table 4. Results on ImageNette for ResNet50 and different input layer large kernel types. CD = Clean Data, OB = OpticsBench, CC = Common corruptions. * = average from multiple seeds. The results on the two corruption benchmarks are averaged across severity and corruption. Bold: best model, underline: second best.

We visualize in Fig. 5 the results on ImageNette for ResNet50 in an absolute fashion to compare the impact of the different kernel classes on corruption accuracy. We plot the baseline and the fully trainable layer (*class II*) together with *class I* and *III* kernel models. The content preserving kernels represent *class I* and the $L1$ prior represents *class III*. Except for color corruptions, both the $L1$ prior model and the fully trainable kernel model help to stabilize the predictions of the baseline model, while the content preserving model helps only for noise and blur at high severities. Interestingly, the $L1$ prior model produces the most favorable results for noise and blur corruptions, followed by the fully trainable kernel, and achieves an accuracy gain of more than 40% at noise severities 4 and 5. The predictions for blur with the $L1$ prior model remain almost constant at 72% accuracy, while the baseline accuracy drops below 40% with increasing severity. The fully trainable kernel model largely stabilizes the predictions, but drops by around 10% at higher severities compared to the $L1$ prior model. The compression corruption type is more challenging for all models compared to the baseline, while the fully trainable kernel and the $L1$ prior models perform similarly and increase in accuracy from severity 3. Interestingly, the baseline also performs quite well here, which may be due to compression artifacts within the original training data. The content preserving kernel model performs significantly worse than the baseline, while the $L1$ prior and the fully trainable kernels tend to slightly increase prediction stability towards higher severities even for hard corruptions (*e.g.* weather). While the $L1$ kernel yields rather high robustness, it is significantly underperforming on clean data.

In summary, the $L1$ prior and the fully trainable versions seem to follow a *similar pattern at similar levels of corruption severities* in Fig. 5 and largely increase the accuracy compared to the baseline for most corruptions and severities. Yet, on clean data (*e.g.* in Table 4) the $L1$ prior leads to significantly lower accuracy (-6.8% w.r.t. the fully trainable filter). In contrast, the content preserving kernel model in Fig. 5 follows a similar trend to the baseline for noise and blur and has the worst performance for all corruptions.

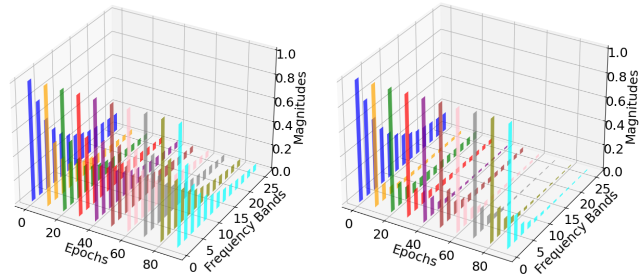
This analysis indicates that for many corruption types, it is beneficial to explicitly encourage the convolutional image-to-image layer to project the input image onto a subspace in which certain spatial frequencies are not represented. The un-regularized fully trainable layer seems to do this implicitly, while preserving the essential information such as to perform well on clean data.

To further deepen our understanding, we present more experiments with other kernels from the three classes in addition to the kernel variants shown in Fig. 5. First, different sizes of the *class I* kernels are compared to see the trade-off between content-preservation and feature locality. The subsequent group analyses different aspects of trainable *class II* filters. The last group compares different *class III* filters to further study the assumed subspace projection.

Content Preserve Kernels (Class I): To have a fully content preserving filter, the kernel needs to have the same size as the to-be-convolved images. Thus, we experiment with two sizes of *class I* kernels. The first kernel has the same size as the input images (225×225). To also be able to compare the *class I* filters with other classes, we designed and trained a 25×25 "content preserving" kernel, *i.e.* a filter that would be content preserving for 25×25 patches. The accuracy of both kernels is given in Table 4. The larger kernel performs significantly worse on the clean data as well as on the common corruptions. However, in the OpticsBench dataset and therefore also in the blur corruptions in Fig. 5, both kernel sizes perform comparably poorly. Filters that purely re-arrange content, whether they preserve locality or not, do not lead to an increase in prediction stability under corruptions.

Fully Trainable Kernels (Class II): We perform two additional experiments: one which replaces our trainable color-dependent (depth-wise) convolution layer with a standard convolution layer of the same kernel size ($25 \times 25 \times 3$). The other uses the proposed depth-wise convolution layer, but with random kernel initialization, which validates the benefit of our fully trainable kernel. Both kernels are *class II* representations.

The standard convolution layer achieves significantly lower accuracy on clean data and both benchmarks. Recombining color channels provides the pre-filtering with more capacity and the ability to better overfit the training data, *i.e.* it provides less generalization. Yet, it yields better results than the *class I* content preserving kernel model.



(a) Without Regularization (b) $L1$ Regularization

Figure 6. Evolution of spectra of (a) Fully Trainable and (b) $L1$ Prior kernels. The bar height indicates the average of the absolute value of the Fourier coefficients in different frequency bands (DC component in the front). Each epoch is normalized separately.

The random initialization model tests initialization of the kernel and achieves comparable results to the fully trainable case for the corrupted data while suffering some loss in peak performance on clean data.

To test the impact of the trainability of our model, we also evaluate two non-trainable blur filters, one rotational-symmetric Gauss blur filter and a directional (horizontal coma) blur filter obtained from OpticsBench [39]. These have both lowpass characteristics and remove high frequency content. From the results in Table 4 the two blur filters have an in-domain accuracy comparable to the fully trainable model. Only the Gaussian blur performs better than the baseline on the two corruption benchmarks, while the directional blur model performs worse. Both perform substantially worse than the fully trained filter.

Projection-type Kernels (Class III): The *class III* kernel ($L1$ prior) in Table 4, tends to reduce the frequencies in the trainable layer via a Lasso regression on the frequency domain of the kernel. This can also be visualized in Fig. 6 (b) where the kernel learns to discard higher frequencies. From Fig. 6 (a), it is evident that the optimization reduces the Fourier coefficients of higher frequencies of freely learned *class II* kernels as well. To check whether numerically small frequency coefficients contribute to the networks' outputs, we use our trained models and remove the low magnitude frequencies without retraining. With this approach, we were able to show that even after removing over 80 % of the frequency information (not necessarily high-frequencies), the model's performance is stable. The results over multiple removal intervals with our models can be examined in Fig. 7. With the insights of Fig. 7 and the increments in performance while training with an $L1$ prior on the frequency domain of the trainable kernel, we conclude that low values in the fully trainable *class II* kernels are numerically zero, *i.e.* they effectively implement projection-type kernels.

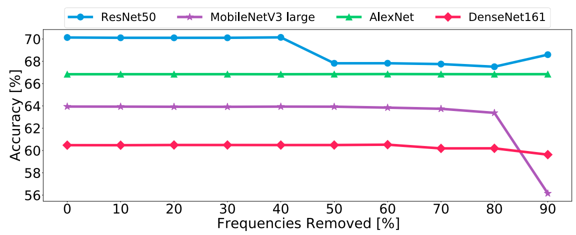


Figure 7. Thresholding the trained convolutional preprocessing kernels in the frequency domain only marginally decreases the accuracy on OpticsBench [39].

5. Discussion

Our main findings can be summarized as follows: a simple convolutional pre-processing layer can significantly improve the robustness against unseen corruptions even when trained only on clean data without dedicated augmentation schemes. An analysis of the learned kernels and experiments with different classes of kernels that were designed to explore different levels of preservation of signal content show that projection-type kernels lead to the most robust results in the majority of corruptions while not significantly reducing peak performance and performance in the case of difficult corruptions (weather and color).

This indicates that a removal of signal content can aid the robustness of classification networks against unknown corruptions with the associated benefits of 1) not having to model expected corruptions for an augmentation-style training and 2) a computationally favorable implementation: only $\approx$ 2000 additional coefficients are needed and training can be performed on a smaller dataset as compared to an augmentation approach.

We observe the usual trade-off of peak-performance vs. robustness. Our experiments indicate that enforcing sparsity of the frequency content of the proposed convolutional pre-processing layer is an alternative way of achieving robust classification results, which is in line with previous findings [63]. While this trade-off also indicates that the signal content responsible for successful classification and possible corruptions do not occupy entirely disparate linear subspaces, it appears as if the signal content responsible for successful classification is essentially a linear subspace of the data. Our proposed convolutional pre-processing layer can therefore be interpreted as being an approximation of the responsible linear subspace. Yet, forcing the projection with the sparsity prior on frequencies yields a shift in the trade-off from high peak-performance to higher robustness. In contrast, an unconstrained non-overcomplete layer can learn to represent the essential content without any prior while better preserving the performance on clean data. A discussion of the relationship to sparse coding is given in the supplementary material.

The complementary experiment of designing signal-content preserving kernels yielded no appreciable performance improvement over the baseline, where the baseline, being an identity transformation, can be interpreted as signal-content preserving as well. This is an indication that the null-space of the high-performance kernels carries information that can lead to over-fitting with an associated decrease in robustness of the classification model.

Our proposed layer could, in principle learn a complete representation of the input images (an identity mapping) without increasing the loss in clean accuracy. The remaining question is why the layer actually learns the subspace projection that allows to focus on more essential information and leads to better generalization. One reason could be that the sparse classes, that are the output of classification, can propagate towards the input layer, paired with the spatial inductive bias of the convolution operation itself, which is trained on image data that is heavily correlated, *i.e.* neighboring pixels tend to be similar. The layer might therefore be biased towards the global image structures and learn to represent fine details only where they are needed to perform well on the training data (*e.g.* the essential high frequency details), which yields the observed benefits. It would be interesting to study why this is not occurring in the standard initial layers of the network, without additional regularization [63]. We hypothesize that this is due to the over-completeness of the representations of most early layers, making it quite likely for a model to represent relevant content as well as noise. In contrast, our simple dimension-preserving layer can at most preserve the input data, and needs gradient signal from the model loss to learn to do so. It therefore learns to predominantly represent signal that is *needed* for the task at hand, *i.e.* the *Essential*.

6. Conclusion & Future Work

We describe a novel, very simple robustifying scheme for classification networks that has the attractive features of being light-weight, both in training and in inference mode, and not requiring knowledge on the corruption model. We therefore believe that this simple technique has a large application potential. However, the current paper is only a first step into its analysis. The convolution proposed and studied in this paper is a space-invariant linear transformation, which appears like a sensible choice for classification problems. An open question is whether other problems like semantic segmentation, tracking, etc. can benefit from similar strategies. In this context, it is further unclear whether space-invariance is a desired property or whether more general linear transformations could be beneficial, *e.g.* in yielding closer approximations to the relevant subspace of the signal content. A connected question is how space-variant processing by the follow-up network affects the initially space-invariant processing by our proposed layer.

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *Proceedings of the European Conference on Computer Vision (ECCV) 2020*, 2020. 2

[2] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*, 2019. 2

[3] David Berthelot, Nicholas Carlini, Ekin D. Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 2

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 2

[5] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Advances in Neural Information Processing Systems*, 2019. 2

[6] Xiangning Chen, Chen Liang, Da Huang, Esteban Real, Kaiyuan Wang, Yao Liu, Hieu Pham, Xuanyi Dong, Thang Luong, Cho-Jui Hsieh, Yifeng Lu, and Quoc V. Le. Symbolic discovery of optimization algorithms, 2023. 1

[7] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. Pali: A jointly-scaled multilingual language-image model, 2023. 3

[8] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11963–11975, 2022. 3

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 1

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the International Conference on Learning Represenations (ICLR)*, 2021. 2, 3

[11] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019. 2

[12] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021. 1

[13] Paul Gavrikov, Janis Keuper, and Margret Keuper. An extended study of human-like behavior under adversarial training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 2361–2368, 2023. 2

[14] 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 *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018. 2

[15] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015. 2, 1

[16] Sven Gowal, Sylvestre-Alvise Rebuffi, Olivia Wiles, Florian Stimberg, Dan Andrei Calian, and Timothy A Mann. Improving robustness using generated data. *Advances in Neural Information Processing Systems*, 34, 2021. 2

[17] Julia Grabinski, Steffen Jung, Janis Keuper, and Margret Keuper. Frequencylowcut pooling–plug & play against catastrophic overfitting. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 2

[18] Julia Grabinski, Janis Keuper, and Margret Keuper. Aliasing and adversarial robust generalization of cnns. *Machine Learning*, pages 1–27, 2022. 2

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 3

[20] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *Proceedings of the International Conference on Learning Represenations (ICLR)*, 2019. 1, 2, 5, 6

[21] Dan Hendrycks*, Norman Mu*, Ekin Dogus Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 1, 2

[22] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021. 2

[23] Robin Hesse, Simone Schaub-Meyer, and Stefan Roth. Content-adaptive downsampling in convolutional neural networks. In *IEEE/CVF Conference on Computer Vision and

Pattern Recognition Workshops (CVPRW), The 6th Efficient Deep Learning for Computer Vision (ECV) Workshop, 2023. 3

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[25] Minui Hong, Jinwoo Choi, and Gunhee Kim. StyleMix: Separating content and style for enhanced data augmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 2

[26] Md Tahmid Hossain, Shyh Wei Teng, Guojun Lu, Mohammad Arifur Rahman, and Ferdous Sohel. Anti-aliasing deep image classifiers using novel depth adaptive blurring and activation function. *Neurocomputing*, 536:164–174, 2023. 2

[27] Jeremy Howard. Imagenette. `https://github.com/fastai/imagenette/`, 2023. 1, 2, 4, 5

[28] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018. 2

[29] Chen Jin, Ryutaro Tanno, Thomy Mertzanidou, Eleftheria Panagiotaki, and Daniel C. Alexander. Learning to downsample for segmentation of ultra-high resolution images. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2022. 3

[30] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009. 1

[31] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2017. 2, 1

[32] Qiufu Li, Linlin Shen, Sheng Guo, and Zhihui Lai. Wavecnet: Wavelet integrated cnns to suppress aliasing effect for noise-robust image classification. *IEEE Transactions on Image Processing*, 30:7074–7089, 2021. 2

[33] Yumeng Li, Dan Zhang, Margret Keuper, and Anna Khoreva. Intra-& extra-source exemplar-based style synthesis for improved domain generalization. *IJCV*, 2023. 2

[34] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022. 3

[35] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, Furu Wei, and Baining Guo. Swin transformer v2: Scaling up capacity and resolution. In *International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 3

[36] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A ConvNet for the 2020s. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11976–11986, 2022. 1, 3

[37] TorchVision maintainers and contributors. TorchVision: PyTorch's Computer Vision library. `https://github.com/pytorch/vision`, 2016. 5

[38] Dmitrii Marin, Zijian He, Peter Vajda, Priyam Chatterjee, Sam Tsai, Fei Yang, and Yuri Boykov. Efficient segmentation: Learning downsampling near semantic boundaries. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019. 3

[39] Patrick Müller, Alexander Braun, and Margret Keuper. Classification robustness to common optical aberrations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 3632–3643, 2023. 1, 2, 5, 6, 7, 8

[40] Chao Peng, Xiangyu Zhang, Gang Yu, Guiming Luo, and Jian Sun. Large kernel matters – improve semantic segmentation by global convolutional network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[41] Rahul Rade and Seyed-Mohsen Moosavi-Dezfooli. Helper-based adversarial training: Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021. 2

[42] Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. In *Advances in Neural Information Processing Systems*, 2021. 3

[43] Sylvestre-Alvise Rebuffi, Sven Gowal, Dan A. Calian, Florian Stimberg, Olivia Wiles, and Timothy Mann. Fixing data augmentation to improve adversarial robustness, 2021. 2

[44] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4322–4330, 2019. 2

[45] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115 (3):211–252, 2015. 4

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3): 211–252, 215. 1

[47] Tonmoy Saikia, Cordelia Schmid, and Thomas Brox. Improving robustness against common corruptions with frequency biased models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10211–10220, 2021. 2

[48] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. ACDC: The adverse conditions dataset with correspondences for semantic driving scene understanding. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10745–10755. IEEE, 2021. 1, 2

[49] Vikash Sehwag, Saeed Mahloujifar, Tinashe Handina, Sihui Dai, Chong Xiang, Mung Chiang, and Prateek Mittal. Improving adversarial robustness using proxy distributions, 2021. 2

[50] Samuel L. Smith, Andrew Brock, Leonard Berrada, and Soham De. ConvNets match vision transformers at scale, 2023. 3

[51] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015. ISSN: 1063-6919. 1

[52] Hossein Talebi and Peyman Milanfar. Learning to resize images for computer vision tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 497–506, 2021. 3

[53] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019. ISSN: 2640-3498. 1

[54] Chittesh Thavamani, Mengtian Li, Nicolas Cebron, and Deva Ramanan. Fovea: Foveated image magnification for autonomous navigation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 15519–15528, 2021. 3

[55] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, pages 776–794. Springer-Verlag, 2020. 2, 4

[56] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-mixer: An all-MLP architecture for vision. In *Advances in Neural Information Processing Systems*, pages 24261–24272. Curran Associates, Inc., 2021. 3

[57] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 32–42, 2021. 1

[58] Zhengzhong Tu, Peyman Milanfar, and Hossein Talebi. Muller: Multilayer laplacian resizer for vision. *ArXiv*, abs/2304.02859, 2023. 3

[59] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020. 2

[60] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*, 2020. 2

[61] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020. 2

[62] Han Xue, Zhiwu Huang, Qianru Sun, Li Song, and Wenjun Zhang. Freestyle layout-to-image synthesis. In *CVPR*, 2023. 2

[63] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, page 13276–13286, Red Hook, NY, USA, 2019. Curran Associates Inc. 1, 2, 8

[64] Jiahui Yu, Zirui Wang, Vijay Vasudevan, Legg Yeung, Mojtaba Seyedhosseini, and Yonghui Wu. Coca: Contrastive captioners are image-text foundation models, 2022. 3

[65] Mehmet Kerim Yucel, Ramazan Gokberk Cinbis, and Pinar Duygulu. Hybridaugment++: Unified frequency spectra perturbations for model robustness. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5718–5728, 2023. 2

[66] Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. CutMix: Regularization strategy to train strong classifiers with localizable features. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6022–6031. IEEE, 2019. 1, 2

[67] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12104–12113, 2022. 3

[68] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *International Conference on Machine Learning*, 2019. 2

[69] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *ICCV*, 2023. 2

[70] Richard Zhang. Making convolutional networks shift-invariant again. In *International conference on machine learning*, pages 7324–7334. PMLR, 2019. 2

[71] Chen Ziwen, Kaushik Patnaik, Shuangfei Zhai, Alvin Wan, Zhile Ren, Alex Schwing, Alex Colburn, and Li Fuxin. Autofocusformer: Image segmentation off the grid. In *CVPR*, 2023. 3

[72] Xueyan Zou, Fanyi Xiao, Zhiding Yu, and Yong Jae Lee. Delving deeper into anti-aliasing in convnets. In *BMVC*, 2020. 2