Improving Feature Stability during Upsampling – Spectral Artifacts and the Importance of Spatial Context

Shashank Agnihotri[1], Julia Grabinski[1,2,3], and Margret Keuper[1,4]

[1] Data and Web Science Group, University of Mannheim
[2] Fraunhofer ITWM, Kaiserslautern
[3] Institute for Machine Learning and Analytics (IMLA), Offenburg University
[4] Max-Planck-Institute for Informatics, Saarland Informatics Campus
{shashank.agnihotri,julia.grabinski,keuper}@uni-mannheim.de

Abstract. Pixel-wise predictions are required in a wide variety of tasks such as image restoration, image segmentation, or disparity estimation. Common models involve several stages of data resampling, in which the resolution of feature maps is first reduced to aggregate information and then increased to generate a high-resolution output. Previous works have shown that resampling operations are subject to artifacts such as aliasing. During downsampling, aliases have been shown to compromise the prediction stability of image classifiers. During upsampling, they have been leveraged to detect generated content. Yet, the effect of aliases during upsampling has not yet been discussed w.r.t. the stability and robustness of pixel-wise predictions. While falling under the same term (*aliasing*), the challenges for correct upsampling in neural networks differ significantly from those during downsampling: when downsampling, some high frequencies can not be correctly represented and have to be removed to avoid aliases. However, when upsampling for pixel-wise predictions, we actually require the model to restore such high frequencies that can not be encoded in lower resolutions. The application of findings from signal processing is therefore a necessary but not a sufficient condition to achieve the desirable output. In contrast, we find that the availability of large spatial context during upsampling allows to provide stable, high-quality pixel-wise predictions, even when fully learning all filter weights.

1 Introduction

Most computer vision models addressing perceptual tasks such as image restoration [16, 89], semantic segmentation [7, 35, 70], optical flow estimation [20, 43, 83] and disparity estimation [6, 11, 50] in realistic scenarios are required to behave in a stable way, at least under mild corruptions. Interestingly, for the slightly simpler task of image classification, recent progress has shown that a model's robustness does not only depend on its training but also on its architecture [29–32, 41, 42, 45, 58, 91, 95]. Specifically, *aliasing*, *i.e.* spectral artifacts that emerge from naïve image resampling, have shown to compromise prediction stability, in particular in the context of classical convolutional models [33, 37, 48, 52, 69, 80, 82] which predominantly use small filter kernels in combination with severe data aggregation during downsampling [30, 58]. Principled cures

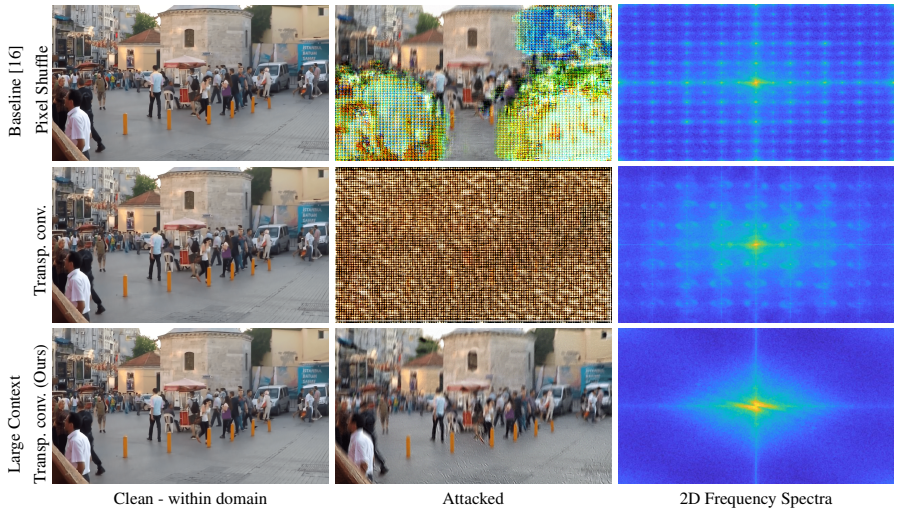

Fig. 1: Image restoration example using NAFNet [16] variants on GoPro [63]. Upsampling techniques like Pixel Shuffle [77] (first row) and transposed convolution [22] using small learnable filters (2×2 or 3×3) (second row) are used by most prior art. Both lead to spectral artifacts for which the model needs to compensate. The clean (in-domain) restored images look appealing - while adversaries (here 5-step PGD [49] attack) can leverage aliases such that artifacts become easily visible. When observed in the frequency domain, they manifest as repeating peaks all over the spectra. Based on sampling theoretic considerations, we propose Large Context Transposed Convolutions (7×7 or larger) (bottom row). They significantly increase the model's stability during upsampling, observable in the restored image under attack and the frequency spectrum.

usually refer to basic concepts from signal processing such as anti-aliasing by blurring before downsampling [29, 91]. While this discussion on classifier (*i.e.* encoder) networks is insightful, it does not provide a recipe to counteract aliases emerging during *upsampling* for pixel-wise prediction tasks such as image restoration. Specifically, naïve upsampling introduces artifacts in the feature representation, such as grid artifacts [4, 65] or ringing artifacts [62]. As shown in Fig. 1, these artifacts, an inherent property of inadequate upsampling (refer Sec. 3) are not always visible to the human eye, are accentuated under adversarial attack such that they can also be seen with a human eye. We leverage this effect in our analysis. When observed in the frequency domain, these artifacts are apparent as multiple peaks, *i.e. aliases* of the original data.

While for downsampling, signal processing laws basically prescribe which part of the information can be retained at lower resolutions without aliases [76], "correct", alias-free upsampling can not restore the original high-resolution information. Thus, learning to upsample feature maps such that the feature stability is not harmed is of paramount importance. In this paper, we therefore first provide a synopsis of different aliases that emerge from different upsampling techniques. Based on this work, we propose a simple, transposed convolution-based upsampling block. We study our proposed operation in the context of various models, from image restoration [16, 89] over semantic segmentation [70] to disparity estimation [50].

Our main contributions can be summarized as follows:

- Motivated by sampling theory [76], we study upsampling in models for diverse pixel-wise prediction tasks. We find that the availability of large kernels in transposed convolutions helps the feature stability and significantly improves over standard, small kernel transposed convolutions as well as pixel shuffle [77].
- While large kernels are required to allow for reduced aliasing and to provide the necessary spatial context for increasing the resolution, additional small kernels can add details and remain useful.
- We provide empirical evidence for our findings on diverse architectures (including vision transformer-based architectures) and downstream tasks such as image restoration, semantic segmentation, and depth estimation.
- We show empirically that our proposed upsampling operation complements other feature stability-increasing approaches like adversarial training.

2 Related Work

In the following, we discuss recent challenges for neural networks regarding artifacts introduced by spatial sampling methods [4, 62, 65]. Further, we review related work on the most recent use of large kernels in CNNs. Finally, we provide an overview of adversarial attacks to gauge the quality of representations learned by a network.

Spectral Artifacts. Several prior works have studied the effect of downsampling operations on model robustness, *e.g.* [2, 29, 30, 42, 46, 91, 95]. Inspired by [30], [29] propose an aliasing-free downsampling in the frequency domain which translates to an infinitely large blurring filter before downsampling in the spatial domain. Thus, for image classification, using large filter kernels has been shown to remove artifacts from downsampled representations and it leads to favorable robustness in all these cases [30, 42, 46]. However, all these works focus on improving the properties of encoder networks.

Models that use transposed convolutions in their decoders[5] are widely used for tasks like image generation [27, 68] or segmentation [7, 55, 64, 70]. However, in simple transposed convolutions, the convolution kernels overlap based on the chosen stride and kernel size. If the stride is smaller than the kernel size, this will cause overlaps in the operation, leading to uneven contributions to different pixels in the upsampled feature map and thus to grid-like artifacts [4, 65]. Further, image resampling can lead to aliases that become visible as ringing artifacts [76]. In the context of deepFake detection, image generation, and deblurring, several works analyzed [14, 18, 21, 23, 38, 44, 47] and improved upsampling techniques [26, 46, 78, 87] to reduce visual artifacts.

Some architectures like PSPNet [93], PSANet [94], or PSMNet [15] simply use bilinear interpolation operations for upsampling the feature representations. While this reduces grid artifacts as bilinear interpolation smoothens out the feature maps, it also has major drawbacks as they sample incorrectly. These new artifacts are sometimes visible as overly smooth predictions, in particular, apparent in the PSPNet segmentation masks. The segmentation masks over-smoothen around edges and often miss out on thin

[5] For more details on Transposed Convolutions refer to [22].

details (predictions showing these are included in the Appendix B.4). This observation already shows why image encoding and decoding have to be considered separately when it comes to sampling artifacts. While during encoding, artifacts can be reduced by blurring, the main purpose of decoder networks is *reducing* blur in many applications, to create fine-granular, pixel-wise accurate outputs, which our approach facilitates.

Large Kernels. For image classification, [54] showed that using large kernels like 7×7 in the CNN convolution layer can outperform self-attention based vision transformers [53, 84]. In [17, 33, 35, 51, 66], the receptive field of the convolution operations was further expanded by using larger kernels, up to 31×31 and 51×51. These larger receptive fields provide more context to the **encoder**, leading to better performance on classification, segmentation, or object detection tasks. [17, 51] use a small kernel in parallel to capture the local context along with the global context. In contrast to these works, which are limited to exploring increased context only during encoding, we investigate if larger kernels can benefit upsampling when considering pixel-wise prediction tasks such as image restoration or segmentation.

Adversarial Attacks. The purpose of adversarial attacks is to reveal neural networks' weaknesses [3, 30, 74, 81] by perturbing pixel values in the input image [12, 28, 49]. These perturbations should lead to a false prediction even though the changes are hardly visible [28, 61, 81]. Especially attacks that have access to the network's architecture and weights, so-called white-box attacks, are a common approach to analyzing weaknesses within the networks' structure [12, 28]. They employ the gradient of the network to optimize the perturbation, which is bounded within an ϵ-ball of the original image, *i.e.* ϵ defines the strength of the attack. Most adversarial attacks are proposed to attack classification networks like the one-step Fast Gradient Sign Method (FGSM) [28] or the multi-step Projected Gradient Descent (PGD) attack [49]. However, they can be adapted to other tasks as *e.g.* in [59, 67, 88]. Furthermore, there are dedicated methods like SegPGD [34] for attacks on semantic segmentation models or PCFA [74] and [71, 73] for optical flow models and CosPGD [3] and others [72] for other pixel-wise prediction tasks. We evaluate the stability of upsampled features using adversarial attacks such as PGD and CosPGD for image restoration and FGSM and SegPGD for segmentation.

3 Spectral Upsampling Artifacts and How They Can Be Reduced

Following, we first theoretically review artifacts that are caused during upsampling from a signal processing aspect. We start by describing the spectral artifacts [76] induced by the bed of nails interpolation, similar to the discussion in [23], and then extend the theoretical analysis to further upsampling schemes. Second, we derive from this analysis two hypotheses for the prediction stability of encoder-decoder networks, depending on their architecture. These hypotheses will motivate the remainder of the manuscript.

Consider, w.l.o.g., a one-dimensional signal I and its discrete Fourier Transform $\mathcal{F}(I)$ with k being the index of discrete frequencies

$$\mathcal{F}(I)_k = \sum_{j=0}^{N-1} e^{-2\pi i \cdot \frac{jk}{N}} \cdot I_j, \quad \text{for} \quad k = 0, \dots, N-1.$$

During decoding, we need to upsample the spatial resolution of I to get I^{up}. For example for an upsampling factor of 2 (often used in DNNs [1, 16, 19, 82, 89]) we have for $\bar{k} = 0, \ldots, 2N-1$

$$\mathcal{F}(I)^{\text{up}}_{\bar{k}} = \sum_{j=0}^{2N-1} e^{-2\pi i \cdot \frac{j\bar{k}}{2 \cdot N}} \cdot I^{\text{up}}_j = \sum_{j=0}^{N-1} e^{-2\pi i \cdot \frac{2 \cdot j\bar{k}}{2 \cdot N}} I_j + \sum_{j=0}^{N-1} e^{-2\pi i \cdot \frac{(2j+1)\bar{k}}{2 \cdot N}} \bar{I}_j, \quad (1)$$

where $\bar{I}_j = 0$ in **bed of nails interpolation**. Therefore, the second term in (1) can be dropped and the first term resembles the original $\mathcal{F}(I)$. Equivalently, we can rewrite Eq. (1), for $\bar{I}_j = 0$, using a Dirac impulse comb as

$$(1) = \sum_{j=0}^{2N-1} e^{-2\pi i \cdot \frac{j\bar{k}}{2 \cdot N}} \cdot \sum_{t=-\infty}^{\infty} I^{\text{up}}_j \cdot \delta(j - 2t). \quad (2)$$

If we now apply the pointwise multiplication with the Dirac impulse comb as convolution in the Fourier domain (assuming periodicity) [25], it is

$$\begin{aligned} \mathcal{F}(I)^{\text{up}}_{\bar{k}} &= \frac{1}{2} \sum_{t=-\infty}^{\infty} \left(\sum_{j=-\infty}^{\infty} e^{-2\pi i \cdot \frac{j\bar{k}}{2N}} I^{\text{up}}_j \right) \left(\bar{k} - \frac{t}{2} \right) \quad (3) \\ &\overset{(1)}{=} \frac{1}{2} \sum_{t=-\infty}^{\infty} \left(\sum_{j=-\infty}^{\infty} e^{-2\pi i \cdot \frac{j\bar{k}}{N}} \cdot I_j \right) \left(\bar{k} - \frac{t}{2} \right) = \frac{1}{2} \sum_{t=-\infty}^{\infty} \mathcal{F}(I)_{\bar{k}} \left(\bar{k} - \frac{t}{2} \right). \end{aligned}$$

We can see that such upsampling creates high-frequency replica of the signal at $\frac{t}{2}$ for t in $-\infty, \ldots, \infty$ in $\mathcal{F}(I)^{\text{up}}$ and spatial frequencies apparent beyond array positions $\frac{N}{2}$ will be impacted by spectral artifacts if no appropriate countermeasures are taken.

A standard countermeasure is interpolation of the inserted values with $\bar{I}_j = \frac{I_{j-1} + I_j}{2}$ for **linear interpolation** in Eq. (1). Linear interpolation (and in consequence bi-linear interpolation in 2D signals) corresponds to a convolution with a triangular impulse with width 2, which can be represented as the convolution of two rectangle functions with width 1. Accordingly, the Fourier response for frequency ℓ, $\mathcal{F}_\ell$ of the triangular impulse is a squared sinc function ($\text{sinc}^2(\ell)$) with $\text{sinc}(\ell) = \frac{\sin(\pi\ell)}{\pi\ell}$. Since the output signal after interpolation is still discrete, i.e. sampled with sampling rate $\frac{1}{2}$, a replica of the interpolation function, the sinc^2 function, will appear with rate 2 in the resulting spectrum (see also Fig. 2). The resulting interpolated signal is not optimal for several reasons. Most importantly, the spectrum of the interpolation function is not flat although the estimated values appear overly smooth (see Fig. 3.). This is arguably suboptimal for, for example, image restoration or segmentation tasks, where fine structural details are supposed to emerge in the upsampled data.

Note that, in Eq. (1), **pixel shuffle** [77] will set $\bar{I}_j$ to completely unrelated values of a different feature map channel, leading to a highly non-smooth signal with frequencies at the band limit. The resulting issues in the spectrum are similar to the ones caused by the bed of nails interpolation. These spectral artifacts can be visually observed in Fig. 3.

Therefore, in **transposed convolutions**, the interpolation function is not fixed to a predefined smoothing kernel but learned so that the resulting signal can represent fine

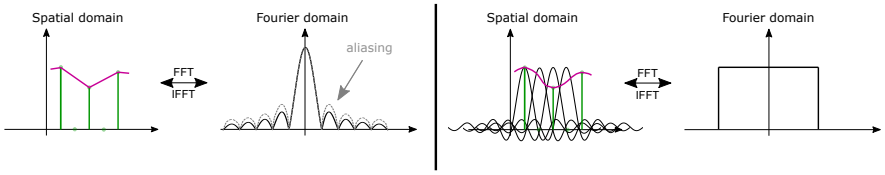

Fig. 2: (Left) Linear interpolation (pink) of the samples (green) causes aliases. (Right) Optimal signal reconstruction (pink) is achieved by sinc interpolation. In practice our spatial context is limited and the interpolation function is discrete. Yet, increasing the kernel size enables the approximation of larger sinc-like structures.

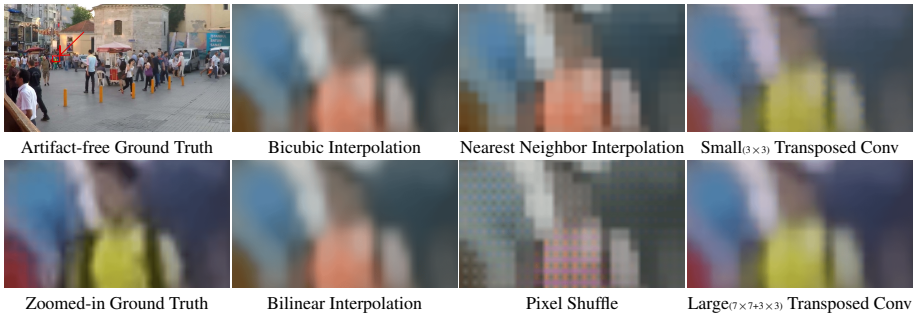

Fig. 3: An image from GoPro [63] downsampled with 3×3 MaxPooling and then upsampled using various upsampling techniques. The resulting artifacts are compared on zoomed-in red box regions for better visibility. Bilinear interpolation causes over-smoothing. Bicubic interpolation causes overestimation along image boundaries while Pixel Shuffle and Nearest Neighbor cause strong grid artifacts along with discoloration. Small kernel transposed convolutions cause grid artifacts, however, on increasing kernel size we start getting better upsampling.

details after the initial bed of nails interpolation and potentially learn to add fine details. One issue is that the learned convolution kernels may overlap based on the chosen stride and kernel size. If the stride is smaller than the kernel size, this will cause overlaps in the operation, leading to uneven contributions to different pixels in the upsampled feature map and thus to grid-like artifacts [4, 65]. Besides this rather technical aspect, transposed convolutions, if sufficiently large (thus also containing more context), could in principle learn to approximate correct upsampling functions. This can be understood when again looking at the Fourier representation. When interpolating, we want to increase the signal array size so that all the original information is preserved and the model can easily learn additional details. Such upsampling to preserve the information from the original low-resolution data is most easily achieved by transforming the signal to the Fourier domain, then padding the missing high-frequency parts with zeros and transforming the resulting array back to the spatial domain [79]. In the Fourier domain, this padding operation can be understood as a point-wise multiplication of the desired full spectrum with a rectangle function with width N (denoted rect_N). Conversely, this operation corresponds to a convolution with $\mathcal{F}^{-1}(\mathrm{rect}_N) = \frac{1}{N}\mathrm{sinc}(xN)$ in the spatial domain. While the $sinc$ function drops off as x increases, it never drops to zero. When applied for interpolation, its crests and the troughs cancel out the aliasing to a large

extent as shown in Fig. 2. Thus, in order to allow the approximation of the optimal interpolation function, the kernel size in transposed convolutions has to be chosen as large as possible. This is, however, at odds with the "learnability" of suitable filter weights. Note that for pixel-wise predictions, models not only need to correctly interpolate, but they also need to "fill in" the missing details, which requires global as well as local context. Therefore, we expect a trade-off on the kernel size of transposed convolutions, where larger kernels improve the stability of the upsampled features and thus can reduce artifacts while the absolute prediction quality can suffer from very large learnable kernels. Sufficiently but not overly large kernels provide sufficient spatial context and are appropriate to allow for the model to learn when to blur and when to preserve/sharpen upsampled features. We illustrate this in Fig. 12 in Appendix C.4.

From this theoretical analysis of common upsampling methods, we derive the following hypotheses that we deem relevant for encoder-decoder architectures:

Hypothesis 1 (H1): ***Large Context Transposed Convolutions (LCTC)*** *i.e. Large kernels in transposed convolution operations provide more context and reduce spectral artifacts and can therefore be leveraged by the network to facilitate better and more robust pixel-wise predictions.*

Hypothesis 2 (H2, Null Hypothesis): *To leverage prediction context and reduce spectral artifacts, it is crucial to increase the size of the* transposed *convolution kernels (upsample using large filters). Increasing the size of normal (i.e. non-upsampling) decoder convolutions does not have this effect.*

In the following, we show the proposed, simple, and principled architecture changes that allow for studying the above hypotheses and improving robustness by improving feature stability.

4 Upsampling using Large Context Transposed Convolutions

Driven by the observations on upsampling artifacts, we investigate the advantage of larger kernel sizes during upsampling, for applications such as semantic segmentation or disparity estimation. Therefore, we keep the models' encoder part fixed and exclusively change operations in the architecture of the decoder part of the model. There, we have two design choices: Upsampling – The kernel size for the transposed convolution operations that learn upsampling, and Decoder Block – The kernel size in the convolution operations of blocks that learn to decode the features. Probing options for Upsampling works towards proving H1 while a combination of both options proves H2, *i.e.* shows that a pure increase in the decoder parameters does not have the desired effect. This is considered in our ablation study in Sec. 5.2.

Figure 4 summarizes the studied options for an abstract encoder-decoder architecture like [70]. The model decoder is depicted in the green box. Operations that we consider to be executed along the red upwards arrows (Upsampling Operators) are detailed in the top right part of the figure (operations a) to c)). Operations that we consider to be executed along the blue sideways arrows (Decoder Building Blocks) are depicted in the bottom right (operations d) to f)).

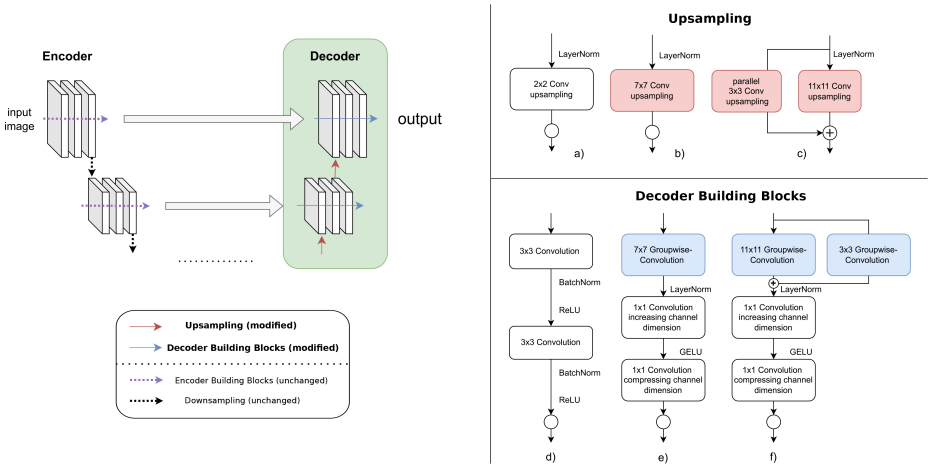

Fig. 4: Abstract representation of an encoder-decoder architecture. While for different tasks, the implementation of the model encoder varies (including transformer-based encoders), our study focuses on the *model decoder* (in green). The backbone for the decoder is commonly a ResNet-like structure for feature extraction [7,70], additionally we also used a ConvNeXt-like [54] structure. We investigate variants of different upsampling operations (the operations along the red arrows in the decoder) for fixed decoder blocks. We consider, as a probe for H1, the baseline transposed deconvolution (a) in the top right), and for LCTC an increased convolution kernel size (b) in the top right), and an increased convolution kernel with a second path using a small convolution kernel (c) in the top right). To test whether the plain increase in parameters is responsible for improved results (zero hypotheses, H2), we also ablate on the increase of convolution kernel size in the decoder block (operations along the blue arrows in the green block), as shown on the bottom right. We consider the common ResNet-like decoder building block structure (in d)) and two ConvNext-like structured backbones for the decoder building block in e) and f), where f) has an additional small convolution applied in parallel, analog to c).

Model Details. Here, we provide details on the studied models. All implementation details are given in the Appendix A.

Transposed Convolution Kernels for Upsampling. The upsampling operation is typically performed with small kernels (2×2 or 3×3) in the transposed convolution operations [8, 13, 70]. We aim to increase the spatial context during upsampling and to reduce grid artifacts. Thus we use **Large Context Transposed Convolutions (LCTC)**. We either use 7×7 transposed convolutions or 11×11 transposed convolutions with a parallel 3×3 transposed convolution. Adding a parallel 3×3 kernel is motivated by [17], as large convolution kernels tend to lose local context, and thus adding a parallel small kernel helps to overcome this potential drawback (see Appendix B.3).

Decoder Building Blocks. To verify that the measurable effects are due to the improved upsampling and not due to merely increasing the decoder capacity, we ablate on decoder convolution blocks similar to convolution blocks used in the ConvNeXt [54] basic block for encoding. While the standard ConvNeXt block uses a 7×7 depth-wise convolution, we consider 7×7 and 11×11 group-wise convolutions, followed by layers present in a ConvNeXt basic block to analyze the importance of the receptive field within the block.

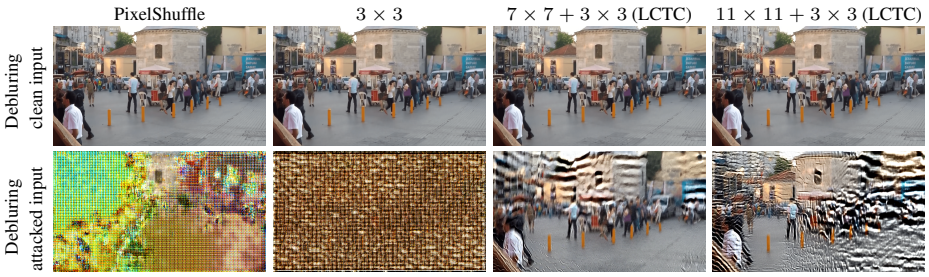

Fig. 5: NAFNet, as proposed, uses Pixel Shuffle for upsampling. We modify only the upsampling operations to transposed convolution with kernel size (3×3) and LCTC (Ours) for comparisons. We observe, for example, under a 10-step PGD attack with $\epsilon \approx \frac{8}{255}$ our proposed H1 gains validity. More examples for [16, 89] using different attacks and budgets are in Appendix C.3.

Table 1: Comparison of performances of different upsampling methods in *SotA* Image Restoration Networks on the GoPro dataset. The architectures use Pixel Shuffle for Upsampling, we propose replacing the Pixel Shuffle with Large Context Transposed Convolutions (LCTC). We report additional results using adversarial training in Tab. 15. Note, that some trade-off between clean performance and robustness is expected [85, 90].

Network	Upsampling Method	Test Accuracy		CosPGD ($\epsilon \approx \frac{8}{255}$) attack iterations						PGD ($\epsilon \approx \frac{8}{255}$) attack iterations					
				5		10		20		5		10		20	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Restormer	Pixel Shuffle	31.99	0.9635	11.36	0.3236	9.05	0.2242	7.59	0.1548	11.41	0.3256	9.04	0.2234	7.58	0.1543
	Transposed Conv 3×3	9.68	0.095	8.24	0.0452	8.53	0.0628	8.44	0.0631	7.66	0.0464	7.72	0.0577	8.64	0.0527
	LCTC: 7×7 + 3 ×3 (Ours)	29.51	0.9337	13.69	0.4186	11.53	0.3136	10.16	0.2484	13.69	0.4183	11.54	0.3137	10.16	0.2483
	LCTC: 11×11 + 3×3 (Ours)	29.44	0.9324	**14.65**	**0.4251**	**12.83**	**0.3438**	**11.48**	**0.29**	**14.65**	**0.4253**	**12.84**	**0.3445**	**11.48**	**0.2893**
NAFNet	Pixel Shuffle	32.87	0.9606	8.67	0.2264	6.68	0.1127	5.81	0.0617	10.27	0.3179	8.66	0.2282	5.95	0.0714
	Transposed Conv 3×3	31.02	0.9422	6.15	0.0332	5.95	0.0258	5.87	0.0233	6.15	0.0332	5.95	0.0258	5.87	0.0234
	LCTC: 7×7 + 3 ×3 (Ours)	31.12	0.9430	**14.54**	**0.4827**	11.05	0.3220	9.06	0.2213	**14.53**	**0.4823**	11.03	0.3201	9.08	0.2224
	LCTC: 11×11 + 3×3 (Ours)	30.77	0.9392	14.34	0.4492	**11.41**	**0.3244**	**9.54**	**0.2411**	14.34	0.45	**11.4**	**0.3236**	**9.55**	**0.2398**

Figure 4 (bottom right e) and f)) shows the structure of a ConvNeXt-style building block used in our work. First, a group-wise convolution is performed, followed by a LayerNorm [5] and two 1×1 convolutions which, similar to [54], creates an inverted bottleneck by first increasing the channel dimension and after a GELU [40] activation compressing the channel dimension again. We consider the ResNet-style building block (Figure 4, d)), with 3×3 convolution, yet without skip connection, as our baseline when studying this architectural design choice.

5 Experiments

In the following, we evaluate the effect of the considered upsampling operators in several applications. We start by evaluating the effect on the upsampled feature stability of recent *state-of-the-art* (SotA) image restoration models [16, 89], then provide results on semantic segmentation using more generic convolutional architectures that allow us to provide compulsory ablations. Last, we show that our results also extend to disparity estimation [50]. We provide details on the used adversarial attacks, datasets, reported metrics, and other experimental details in Appendix A.

In all cases, we observe that Large Context Transposed Convolutions (LCTC) improve the results of the respective pixel-wise prediction task in terms of stability under attack, showing that H1 holds. Further, our extensive ablation on image segmentation shows that increasing the convolution kernel in the decoder building blocks does not have this beneficial effect, providing experimental evidence for our hypothesis H2 and confirming the impact of spectral artifacts on pixel-wise predictions.

5.1 Image Restoration

For image restoration, we consider the Vision Transformer-based Restormer [89] and NAFNet [16]. Both originally use the Pixel Shuffle [77] for upsampling. Here, we compare the reconstructions from these proposed architectures to their variants using the proposed operators with large transposed convolution filters. We use the same metrics as [16, 89], Peak Signal-to-Noise Ratio (PSNR), and structural similarity index measure (SSIM) [86]. We perform our experiments on the GoPro [63] image deblurring dataset, following the experimental setup in [1].

Results on Image Restoration. We first consider qualitative results on NAFNet [16] in Figure 5 and Restormer [89] in Fig. 10, Fig. 11(in Appendix C.3), where we see that the proposed upsampling operators allow for visually good results in image deblurring on clean data (similar to pixel shuffle). Yet, in contrast to pixel shuffle and the baseline small transposed convolution filters, the proposed Large Context Transposed Convolutions (LCTC) significantly reduces artifacts that arise on attacked images (in this case, 10-step PGD with $\epsilon \approx \frac{8}{255}$). attacks with varying numbers of steps.

In Table 1, we report the average PSNR and SSIM values of the reconstructed images from the GoPro test set. These results confirm that at filter size 3×3, the performance of the transposed convolution variant of both the considered networks is significantly worse than the originally proposed Pixel Shuffle variant, justifying the community's extensive use of Pixel Shuffle. However, we observe on increasing context by increasing the kernel size to 7×7 that the performance of the transposed convolution variants significantly improves, especially making the networks more stable when facing adversarial attacks. This boost in performance is further accentuated by increasing the kernel size to 11×11 (both with parallel small kernels). These results provide evidence for Hypothesis 1.

Note that the slightly reduced performance on clean images, seen in Table 1, is expected to some degree: here, we only investigate sampling in the decoder, while pixel unshuffle is used in the encoder, potentially causing a mismatch. Further, previous works have shown that there exists a trade-off between adversarial robustness and clean performance [85, 90]. However, we do not observe this trade-off for matching encoder-decoder architectures, *e.g.* in semantic segmentation.

5.2 Semantic Segmentation

As baseline architecture for semantic segmentation, we consider a UNet-like architecture [70] with encoder backbone layers from ConvNeXt [54] (see Appendix B.2 on the choice of encoder). This generic architecture facilitates providing a thorough ablation

Fig. 6: A comparison of semantic segmentation mask predictions for the shown input images. The row labeled "Prediction Difference" shows the difference in predictions between the baseline model and the model with Large Context Transposed Convolutions (11×11+3×3 kernels). On white pixels, both models agree. Red pixels indicate that the baseline model predicts correctly but our modified model predicts incorrectly. Green pixels indicate that our modified model predicts correctly but the baseline does not. The ground truth segmentation boundaries are drawn in black. Our modification improves the segmentation result along object boundaries, which can be attributed to spectral artifact removal, but also in more extended regions, where the context plays a more crucial role.

Table 2: Semantic Segmentation performance on the PASCAL VOC2012 validation set for UNet with ConvNeXt encoder, and the baseline UNet decoder (see Figure 4) with differently sized kernels in transposed convolution for feature map upscaling while keeping rest of the architecture fixed. Additional results are provided in Tab. 7 and Tab. 8 in Appendix B.1.

Transposed Convolution Kernels	**Clean Test Accuracy**			**FGSM attack epsilon** $\frac{1}{255}$			**FGSM attack epsilon** $\frac{8}{255}$			**SegPGD** ($\epsilon \approx \frac{8}{255}$) **attack iterations** 3			**SegPGD** ($\epsilon \approx \frac{8}{255}$) **attack iterations** 20		
	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc
2×2 (baseline)	78.34	86.89	95.15	53.54	70.96	86.08	47.02	65.41	82.78	23.06	46.51	45.30	5.54	18.79	23.72
LCTC: 7×7 (Ours)	78.92	**88.06**	95.23	56.02	74.13	86.45	49.24	68.89	82.87	26.53	53.05	61.16	**7.17**	23.05	**27.52**
LCTC: 11×11 + 3×3 (Ours)	**79.33**	87.81	**95.41**	**58.04**	**74.93**	**87.80**	**51.25**	**69.31**	**84.64**	**27.49**	**53.08**	**64.13**	7.08	**23.30**	26.82

on all considered blocks in the decoder network. Our experiments are conducted on the PASCAL VOC 2012 dataset [24]. We report the mean Intersection over Union (mIoU) of the predicted and ground truth segmentation mask, the mean accuracy over all pixels (mAcc), and the mean accuracy over all classes (allAcc).

Results on Semantic Segmentation. We first discuss the results for different upsampling operations. The remaining architecture is kept identical, with ResNet-style building blocks in the decoder, throughout these experiments. The clean test accuracies are shown in Table 2. We see that as we increase the kernel size of the transposed convolution layers, there is a slight increase across all three evaluation metrics. Moreover, Figure 6 visually demonstrates that, as we increase the size of the kernels in transposed convolution from 2×2 (baseline) to 11×11, the segmentations of the thin end and protrusions, for example, in the wing of the aircraft sample image are improving. The baseline model with small transposed convolution kernels could not predict these details. As hypothesized in H1, we observe that increasing the context can reduce spectral artifacts caused when representation and images are upsampled using LCTC.

Further, in Table 2, we evaluate the performance of the segmentation models against FGSM [28] and the multi-step attack SegPGD [34] adversarial attacks for the indicated ϵ values. As expected, with the increasing intensity of the attack, the performance of all models drops. Yet, even at high attack intensities, the larger kernels perform better than

Table 3: Adversarially trained models using FGSM ($\epsilon \approx \frac{8}{255}$) from Table 2 tested against SegPGD adversarial attacks ($\epsilon \approx \frac{8}{255}$) on UNet with ConvNeXt encoder and decoder with different sized kernels in the transposed convolution for upsampling, while keeping rest of the architecture identical. See Tab. 10 in Appendix B.6 for more evaluations including PGD training.

Transposed Convolution Kernels	**Clean Test Data**			**SegPGD attack iterations** 3			20		
	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc
2×2	78.57	86.68	95.23	26.59	48.99	67.71	7.6	24.06	31.37
LCTC: 7×7 (Ours)	78.41	86.22	95.20	28.11	53.39	66.30	8.36	28.54	28.13
LCTC: 11×11 + 3×3 (Ours)	**79.57**	**88.1**	**95.3**	**30.37**	**55.54**	**68.3**	**9.4**	**29.79**	**32.37**

Table 4: Empirical evaluations for H2 using a UNet with ConvNeXt encoder. We observe that across different-sized kernels in transposed convolution, for a fixed kernel size, increasing the context in the decoder building blocks by using larger kernels causes performance deterioration. These observations for image decoding contrast the findings on image encoding by [17, 51, 54].

Transposed Convolution Kernels	**Decoder Building Block Style**	**Test Accuracy**	**FGSM attack epsilon** $\frac{1}{255}$	$\frac{8}{255}$	**SegPGD** ($\epsilon \approx \frac{8}{255}$) **attack iterations** 3	20
		mIoU / mAcc / allAcc	mIoU / mAcc / allAcc	mIoU / mAcc / allAcc	mIoU / mAcc / allAcc	mIoU / mAcc / allAcc
2×2	ResNet Style 3×3	78.34 / 86.89 / 95.15	53.54 / 70.96 / 86.08	47.02 / 65.41 / 82.78	23.06 / 46.51 / 60.04	5.54 / 18.79 / 23.72
	ConvNeXt style 7×7	77.17 / 86.86 / 94.81	49.98 / 72.22 / 83.93	42.04 / 64.86 / 79.08	17.94 / 44.81 / 47.96	3.20 / 14.73 / 9.81
	ConvNeXt style 11×11 + 3×3	77.17 / 86.86 / 94.81	47.34 / 67.72 / 83.34	37.91 / 57.79 / 78.21	13.97 / 35.82 / 45.68	2.21 / 10.75 / 5.29
LCTC: 7×7 (Ours)	**ResNet Style 3×3**	78.92 / **88.06** / 95.23	56.02 / 74.13 / 86.45	49.24 / 68.89 / 82.87	26.53 / 53.05 / 61.16	**7.17** / 23.05 / **27.52**
	ConvNeXt style 7×7	77.57 / 87.04 / 94.92	52.93 / 72.18 / 85.51	44.89 / 65.71 / 80.74	17.64 / 43.32 / 47.80	1.86 / 7.18 / 3.55
	ConvNeXt style 11×11 + 3×3	77.99 / 87.86 / 94.96	51.61 / 73.01 / 84.85	43.93 / 66.22 / 80.73	17.07 / 42.30 / 48.78	1.80 / 7.11 / 3.04
LCTC: 11×11 +3×3 (Ours)	**ResNet Style 3×3**	**79.33** / 87.81 / **95.41**	**58.04 / 74.93 / 87.80**	**51.25 / 69.31 / 84.64**	**27.49 / 53.08 / 64.13**	7.08 / **23.30** / 26.82
	ConvNeXt style 7×7	78.32 / 86.98 / 95.09	53.31 / 72.45 / 86.16	44.89 / 65.18 / 82.03	16.14 / 40.65 / 50.39	1.93 / 9.35 / 3.90
	ConvNeXt style 11×11 + 3×3	77.42 / 86.24 / 94.94	54.48 / 72.53 / 86.25	46.67 / 66.59 / 82.29	18.76 / 44.60 / 51.49	2.31 / 8.70 / 3.50

the small ones, and we see a trend of improvement in performance as we increase the kernel size, providing more evidence for Hypothesis 1.

Ablation Study. In the following, we first consider the effects of additional adversarial training, then ablate on the impact of other decoder building blocks and the filter size. Variations of the model encoder are ablated in the Appendix B.2, the impact of using small parallel kernels in addition to large kernels is ablated and discussed in Appendix B.3, and competing upsampling techniques are ablated in Appendix B.5.

Adversarial Training. In Table 3, we report results for FGSM adversarially trained models under SegPGD attack, with attacks as in Table 2. While the overall performance under attack is improved as expected, the trend of LCTC providing better results persists. More results for FGSM attack and SegPGD attacks with different numbers of iterations are given in Tab. 7 and Tab. 8 in the Appendix. In Table 15, we additionally evaluate image restoration models under adversarial training.

Change in the decoder backbone architecture. While all previous experiments focused on the upsampling using transposed convolutions in the decoder, we now evaluate the influence of the convolutional kernel size within the decoder which does not upsample (see Section 4). For these experiments, we use a UNet-like architecture with a ConvNeXt backbone in the encoder and the PASCAL VOC 2012 dataset.

In Table 4 we observe, for a fixed transposed convolution kernel size, as we increase the size of the convolution kernel in the decoder building blocks, the performance of the model *decreases*. This phenomenon extends to the performance of the architectures

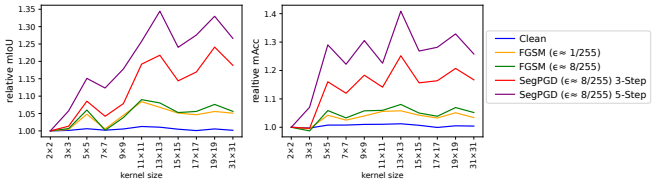

Fig. 7: Performance comparison on PASCAL VOC2012 using UNet with ConvNeXt encoder for different LCTC sizes from 2×2 (small) to 31×31 (LCTC) kernels. All, besides the baseline with 2×2 and 3×3, have a parallel 3×3 kernel, as shown in Figure 4 (bottom left). For the decoder building block backbone, a ResNet Style 3×3 style is used. See Tab. 9 for the values.

Table 5: Comparison of performances of different upsampling methods in the UNet-like architecture. All architectures use the baseline (ConvNeXt) encoder and 3×3 convolution kernels in the decoder block. Please refer to Table 13 in Appendix B.5 for more evaluations and discussion, including those with ConvNeXt style 7×7+3×3 Convolution kernels in the decoder blocks.

Upsampling Method	**Test Accuracy**			**FGSM attack epsilon**						**SegPGD** ($\epsilon \approx \frac{8}{255}$) **attack iterations**					
				$\frac{1}{255}$			$\frac{8}{255}$			5			20		
	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc	mIoU	mAcc	allAcc
Pixel Shuffle	78.54	87.32	95.18	53.82	71.58	85.88	46.67	65.03	81.71	15.06	38.85	41.71	6.69	23.43	24.05
Nearest Neighbour Interpolation	78.40	**88.16**	95.09	52.68	73.51	84.55	46.08	67.96	80.22	15.34	**44.53**	36.21	**7.65**	**27.89**	20.48
Transposed Convolution 2×2	78.45	86.66	95.20	53.76	70.62	86.32	47.33	64.58	83.16	14.43	35.50	45.30	5.54	18.79	23.72
LCTC: 11×11+3×3 (Ours)	**79.33**	87.81	**95.41**	**58.04**	**74.93**	**87.80**	**51.25**	**69.31**	**84.64**	**18.15**	43.51	**49.36**	7.08	23.30	**26.82**

under adversarial attacks, showing that a mere increase in parameters in the model decoder does not have a positive effect on model performance or on its stability. This proves the validity of hypothesis H2. An explanation for this phenomenon could be that we only need to increase context during the actual upsampling step, increasing context in the consequent decoder building blocks has a negligible effect on the quality of representations learned. However, the increase in the number of parameters makes the architecture more susceptible to adversarial attacks.

Ablation on filter size saturation. After proving H1 one could argue that networks will consistently improve with increased kernel size for Large Context Transposed Convolutions. Hence, we test larger kernel sizes of 15×15, 17×17, 19×19 and 31×31 kernels. Yet, as seen in Figure 7, the effect of the kernel size appears to saturate: the performance after 13×13 and the performance of 31×31 kernels is not better than for 11×11 kernels. Yet, they are significantly better than the baseline's performance.

Ablation on different Upsampling Methods. Following, we compare different upsampling techniques thus justifying our advocacy for using LCTC instead of other upsampling techniques like interpolation and pixel shuffle in the real world.

We report the comparison in Table 5 and observe that both Pixel shuffle and Nearest Neighbor interpolation perform better than the usually used Transposed Convolution with a 2×2 kernel size. However, as we increase the kernel size for Transposed Convolution to 11×11 with a 3×3 small kernel in parallel, we observe that LCTC is strictly outperforming Pixel Shuffle, on both clean unperturbed images and under adversarial attacks, across all metrics used. Large Context Transposed Convolutions are either outperforming or performing at par with Nearest Neighbor interpolation. Thus we demonstrate the superior clean and adversarial performance of Large Context Transposed Convolutions operation over other commonly used techniques.

5.3 Disparity Estimation

To show that the observations extend from image restorations and segmentation to other tasks, we conduct additional experiments for disparity estimation. We consider the STTR-light [50] architecture, built from STTR, which is a recent state-of-the-art vision-transformer based model for disparity estimation and occlusion detection. To implement the proposed modification, we alter the kernel sizes in the transposed convolution layers used for pixel-wise upsampling in the "feature extractor" module of the architecture from 3×3 kernels to larger kernels. We conduct evaluations on FlyingThings3D [60] and keep all other details as implemented in [50].

In Table 6, we report the improvements in performance due to our architecture modification of increasing the size of the transposed convolution kernels used for upsampling, from the 3×3 in the baseline model to 7×7 (LCTC). Similar to previous applications, the increased kernel sizes with parallel 3×3 kernels further facilitate to stabilize the model when attacked, as evaluated here for 3 attack iterations using PGD with $\epsilon \approx \frac{8}{255}$ on the disparity loss. Indicating that larger kernels in the transposed convolutions can better decode learned representations from the encoder regardless of the specific downstream task. We provide visual results in Appendix D.1.

Table 6: Comparison of performance of STTR-light architecture with different sized kernels in transposed convolution for upsampling the feature maps in the feature extractor (lower is better). The entire set of results is provided as Tab. 17 in Appendix D.1.

Transposed Convolution Kernels	**Test Accuracy**		**3-Step PGD attack**	
	epe↓	3px error↓	epe↓	3px error↓
STTR-light [50] reported	0.5	1.54	-	-
3×3 [50] reproduced	0.4927	1.54	4.05	18.5
LCTC: 7×7 + 3×3 (Ours)	**0.4788**	**1.50**	**4.02**	**18.3**

6 Conclusion

We provide conclusive reasoning and empirical evidence for our hypotheses on the importance of context during upsampling. While increasing the size of convolutions during upsampling (LCTC) increases prediction stability, increasing the size of those convolution layers without upsampling does not benefit the network. This indicates that observations made for increased context during encoding do not translate to decoding. Further, we show that our simple LCTC can be directly incorporated into recent models, yielding better stability even in ViT-based architectures like Restormer, NAFNet, and STTR-light as well as in classical CNNs. Our observations are consistent across several architectures and downstream tasks.

Limitations. Current metrics for measuring performance do not completely account for spectral artifacts. Spectral artifacts begin affecting these metrics only when they become pronounced such as under adversarial attacks, and here LCTC consistently performs better across tasks and architectures. Ideally, we would want infinitely large kernels, however, with increasing kernel size and task complexity, training extremely large kernels can be challenging. Thus, in this work, while having ablated over kernels as large as 31×31, we propose using kernels only as large as 7×7 to 11×11 for good practical trade-offs. Further improvements might be possible when jointly optimizing the encoder *and* decoder. Moreover, there might exist other factors that contribute to the introduction and existence of spectral artifacts such as spatial bias.

Acknowledgements.

Margret Keuper acknowledges funding by the DFG Research Unit 5336 - Learning to Sense. The OMNI cluster of the University of Siegen was used for some of the initial computations. Additionally, Shashank Agnihotri would like to thank Dr. Bin Zhao for his help in translating [75].

References

1. Agnihotri, S., Gandikota, K.V., Grabinski, J., Chandramouli, P., Keuper, M.: On the unreasonable vulnerability of transformers for image restoration-and an easy fix. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3707–3717 (2023)
2. Agnihotri, S., Grabinski, J., Keuper, J., Keuper, M.: Beware of aliases–signal preservation is crucial for robust image restoration. arXiv preprint arXiv:2406.07435 (2024)
3. Agnihotri, S., Jung, S., Keuper, M.: CosPGD: A unified white-box adversarial attack for pixel-wise prediction tasks. In: International Conference on Machine Learning (2024)
4. Aitken, A., Ledig, C., Theis, L., Caballero, J., Wang, Z., Shi, W.: Checkerboard artifact free sub-pixel convolution: A note on sub-pixel convolution, resize convolution and convolution resize. arXiv preprint arXiv:1707.02937 (2017)
5. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016)
6. Badki, A., Troccoli, A., Kim, K., Kautz, J., Sen, P., Gallo, O.: Bi3D: Stereo depth estimation via binary classifications. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2020)
7. Badrinarayanan, V., Kendall, A., Cipolla, R.: Segnet: A deep convolutional encoder-decoder architecture for image segmentation. IEEE transactions on pattern analysis and machine intelligence **39**(12), 2481–2495 (2017)
8. Baranchuk, D., Rubachev, I., Voynov, A., Khrulkov, V., Babenko, A.: Label-efficient semantic segmentation with diffusion models (2021)
9. Brigham, E.O., Morrow, R.: The fast fourier transform. IEEE spectrum **4**(12), 63–70 (1967)
10. Brunton, S.L., Kutz, J.N.: Data-driven science and engineering: Machine learning, dynamical systems, and control. Cambridge University Press (2022)
11. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: European conference on computer vision. pp. 213–229. Springer (2020)
12. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: 2017 ieee symposium on security and privacy (sp). pp. 39–57. IEEE (2017)
13. Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., Joulin, A.: Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems **33**, 9912–9924 (2020)
14. Chandrasegaran, K., Tran, N.T., Cheung, N.M.: A closer look at fourier spectrum discrepancies for cnn-generated images detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 7200–7209 (June 2021)
15. Chang, J.R., Chen, Y.S.: Pyramid stereo matching network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 5410–5418 (2018)
16. Chen, L., Chu, X., Zhang, X., Sun, J.: Simple baselines for image restoration. In: European Conference on Computer Vision. pp. 17–33. Springer (2022)
17. Ding, X., Zhang, X., Han, J., Ding, G.: 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. pp. 11963–11975 (2022)

18. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. Advances in neural information processing systems **29** (2016)
19. Dosovitskiy, A., Brox, T.: Inverting visual representations with convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4829–4837 (2016)
20. Dosovitskiy, A., Fischer, P., Ilg, E., Hausser, P., Hazirbas, C., Golkov, V., Van Der Smagt, P., Cremers, D., Brox, T.: Flownet: Learning optical flow with convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2758–2766 (2015)
21. Dosovitskiy, A., Springenberg, J.T., Tatarchenko, M., Brox, T.: Learning to generate chairs, tables and cars with convolutional networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **39**(4), 692–705 (2017)
22. Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285 (2016)
23. Durall, R., Keuper, M., Keuper, J.: Watch your up-convolution: Cnn based generative deep neural networks are failing to reproduce spectral distributions. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 7890–7899 (2020)
24. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html (2012)
25. Forsyth, D.A., Ponce, J.: Computer vision: a modern approach. Prentice Hall professional technical reference (2002)
26. Gal, R., Hochberg, D.C., Bermano, A., Cohen-Or, D.: SWAGAN: A style-based wavelet-driven generative model. ACM Transactions on Graphics (TOG) **40**(4), 1–11 (2021)
27. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
28. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572 (2014)
29. Grabinski, J., Jung, S., Keuper, J., Keuper, M.: Frequencylowcut pooling-plug and play against catastrophic overfitting. In: European Conference on Computer Vision. pp. 36–57. Springer (2022)
30. Grabinski, J., Keuper, J., Keuper, M.: Aliasing and adversarial robust generalization of cnns. Machine Learning pp. 1–27 (2022)
31. Grabinski, J., Keuper, J., Keuper, M.: Aliasing coincides with cnns vulnerability towards adversarial attacks. In: The AAAI-22 Workshop on Adversarial Machine Learning and Beyond. pp. 1–5 (2022)
32. Grabinski, J., Keuper, J., Keuper, M.: Fix your downsampling asap! be natively more robust via aliasing and spectral artifact free pooling (2023)
33. Grabinski, J., Keuper, J., Keuper, M.: As large as it gets – studying infinitely large convolutions via neural implicit frequency filters. Transactions on Machine Learning Research (2024), https://openreview.net/forum?id=xRy1YRcHWj, featured Certification
34. Gu, J., Zhao, H., Tresp, V., Torr, P.H.: Segpgd: An effective and efficient adversarial attack for evaluating and boosting segmentation robustness. In: European Conference on Computer Vision. pp. 308–325. Springer (2022)
35. Guo, M.H., Lu, C.Z., Hou, Q., Liu, Z., Cheng, M.M., Hu, S.M.: Segnext: Rethinking convolutional attention design for semantic segmentation. Advances in Neural Information Processing Systems **35**, 1140–1156 (2022)
36. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: 2011 international conference on computer vision. pp. 991–998. IEEE (2011)

37. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
38. He, Y., Yu, N., Keuper, M., Fritz, M.: Beyond the spectrum: Detecting deepfakes via resynthesis. In: Zhou, Z.H. (ed.) Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21. pp. 2534–2541. International Joint Conferences on Artificial Intelligence Organization (8 2021), main Track
39. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. Proceedings of the International Conference on Learning Representations (2019)
40. Hendrycks, D., Gimpel, K.: Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415 (2016)
41. Hoffmann, J., Agnihotri, S., Saikia, T., Brox, T.: Towards improving robustness of compressed cnns. In: ICML Workshop on Uncertainty and Robustness in Deep Learning (UDL) (2021)
42. Hossain, M.T., Teng, S.W., Lu, G., Rahman, M.A., Sohel, F.: Anti-aliasing deep image classifiers using novel depth adaptive blurring and activation function. Neurocomputing **536**, 164–174 (2023)
43. Ilg, E., Mayer, N., Saikia, T., Keuper, M., Dosovitskiy, A., Brox, T.: Flownet 2.0: Evolution of optical flow estimation with deep networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2462–2470 (2017)
44. Jung, S., Keuper, M.: Spectral distribution aware image generation. In: Proceedings of the AAAI conference on artificial intelligence. vol. 35, pp. 1734–1742 (2021)
45. Jung, S., Lukasik, J., Keuper, M.: Neural architecture design and robustness: A dataset. In: Eleventh International Conference on Learning Representations. OpenReview.net (2023)
46. Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., Aila, T.: Alias-free generative adversarial networks. Advances in Neural Information Processing Systems **34**, 852–863 (2021)
47. Khayatkhoei, M., Elgammal, A.: Spatial frequency bias in convolutional generative adversarial networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 36, pp. 7152–7159 (2022)
48. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems **25** (2012)
49. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial machine learning at scale. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=BJm4T4Kgx
50. Li, Z., Liu, X., Drenkow, N., Ding, A., Creighton, F.X., Taylor, R.H., Unberath, M.: Revisiting stereo depth estimation from a sequence-to-sequence perspective with transformers. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 6197–6206 (2021)
51. Liu, S., Chen, T., Chen, X., Chen, X., Xiao, Q., Wu, B., Kärkkäinen, T., Pechenizkiy, M., Mocanu, D., Wang, Z.: More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. arXiv preprint arXiv:2207.03620 (2022)
52. Liu, S., Deng, W.: Very deep convolutional neural network based image classification using small training sample size. In: 2015 3rd IAPR Asian conference on pattern recognition (ACPR). pp. 730–734. IEEE (2015)
53. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (2021)

54. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11976–11986 (2022)
55. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3431–3440 (2015)
56. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: International Conference on Learning Representations (2017), https://openreview.net/forum?id=Skq89Scxx
57. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bkg6RiCqY7
58. Maiya, S.R., Ehrlich, M., Agarwal, V., Lim, S.N., Goldstein, T., Shrivastava, A.: A frequency perspective of adversarial robustness (2021)
59. Mathew, A., Patra, A., Mathew, J.: Monocular depth estimators: Vulnerabilities and attacks. ArXiv **abs/2005.14302** (2020)
60. Mayer, N., Ilg, E., Hausser, P., Fischer, P., Cremers, D., Dosovitskiy, A., Brox, T.: A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4040–4048 (2016)
61. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2574–2582 (2016)
62. Mosleh, A., Langlois, J.M.P., Green, P.: Image deconvolution ringing artifact detection and removal via psf frequency analysis. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. pp. 247–262. Springer International Publishing, Cham (2014)
63. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (July 2017)
64. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision. pp. 1520–1528 (2015)
65. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill **1**(10), e3 (2016)
66. Peng, C., Zhang, X., Yu, G., Luo, G., Sun, J.: Large kernel matters – improve semantic segmentation by global convolutional network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
67. Pervin, M., Tao, L., Huq, A., He, Z., Huo, L., et al.: Adversarial attack driven data augmentation for accurate and robust medical image segmentation. arXiv preprint arXiv:2105.12106 (2021)
68. Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. arXiv preprint arXiv:1511.06434 (2015)
69. Radosavovic, I., Kosaraju, R.P., Girshick, R., He, K., Dollar, P.: Designing network design spaces. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2020)
70. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18. pp. 234–241. Springer (2015)
71. Scheurer, E., Schmalfuss, J., Lis, A., Bruhn, A.: Detection defenses: An empty promise against adversarial patch attacks on optical flow. arXiv preprint arXiv:2310.17403 (2023)

72. Schmalfuss, J., Mehl, L., Bruhn, A.: Attacking motion estimation with adversarial snow. arXiv preprint arXiv:2210.11242 (2022)
73. Schmalfuss, J., Mehl, L., Bruhn, A.: Distracting downpour: Adversarial weather attacks for motion estimation (2023)
74. Schmalfuss, J., Scholze, P., Bruhn, A.: A perturbation-constrained adversarial attack for evaluating the robustness of optical flow (2022)
75. segcv: segcv/pspnet. https://github.com/segcv/PSPNet/blob/master/Train.md (2021)
76. Shannon, C.E.: Communication in the presence of noise. Proceedings of the IRE **37**(1), 10–21 (1949)
77. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network (2016)
78. Si-Yao, L., Ren, D., Yin, Q.: Understanding kernel size in blind deconvolution. In: 2019 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 2068–2076. IEEE (2019)
79. Smith III, J.O.: Physical audio signal processing: For virtual musical instruments and audio effects. (No Title) (2010)
80. Sommerhoff, H., Agnihotri, S., Saleh, M., Moeller, M., Keuper, M., Kolb, A.: Differentiable sensor layouts for end-to-end learning of task-specific camera parameters. arXiv preprint arXiv:2304.14736 (2023)
81. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
82. Tan, M., Le, Q.: Efficientnetv2: Smaller models and faster training. In: International conference on machine learning. pp. 10096–10106. PMLR (2021)
83. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. pp. 402–419. Springer (2020)
84. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)
85. Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A.: Robustness may be at odds with accuracy. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=SyxAb30cY7
86. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing **13**(4), 600–612 (2004)
87. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. Advances in neural information processing systems **27** (2014)
88. Yamanaka, K., Matsumoto, R., Takahashi, K., Fujii, T.: Adversarial patch attacks on monocular depth estimation networks. IEEE Access **8**, 179094–179104 (2020)
89. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5728–5739 (2022)
90. Zhang, H., Yu, Y., Jiao, J., Xing, E., El Ghaoui, L., Jordan, M.: Theoretically principled trade-off between robustness and accuracy. In: ICML (2019)
91. Zhang, R.: Making convolutional networks shift-invariant again. In: ICML (2019)
92. Zhao, H.: semseg. https://github.com/hszhao/semseg (2019)
93. Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J.: Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2881–2890 (2017)

94. Zhao, H., Zhang, Y., Liu, S., Shi, J., Loy, C.C., Lin, D., Jia, J.: PSANet: Point-wise spatial attention network for scene parsing. In: ECCV (2018)
95. Zou, X., Xiao, F., Yu, Z., Lee, Y.J.: Delving deeper into anti-aliasing in convnets. In: BMVC (2020)