Reverse Convolution and Its Applications to Image Restoration

Supplementary Material

In this supplementary material, we provide additional details and analyses of our proposed reverse convolution operator and block. First, we explore its integration with the widely used UNet architecture in Section A. To further demonstrate the effectiveness of the proposed operator, we apply it to the classical deblurring task in Section B. Additionally, to highlight the unique characteristics of the reverse convolution operator in comparison to other methods, we conduct experiments on a single-layer network utilizing the operator in Section C. And to better explain the details of our proposed operator, we present a detailed description of the datasets and training process in Section E and derive Eq. (2) in Section F. Moreover, to further evaluate the effectiveness and flexibility of our operator, we integrate it into several advanced architectures, including ResNet, DenseNet, and FCN, to test its performance on high-level vision tasks, as discussed in Section G. Finally, we analyze some visual details, including the kernel activation patterns and λ features, in Section H.

A. Denoising with UNet architecture

It is well-known that UNet [12] is effective for image-to-image translation tasks such as image denoising. In this experiment, we replace the basic convolution operations of plain UNet with our proposed converse block, which we refer to as Converse-UNet (see Figure 1). The total number of converse block is set to 20. We compare two variants with depthwise convolution and depthwise transposed convolution, which we refer to as Conv-UNet and ConvT-UNet, respectively. The number of parameters and average PSNR(dB) results of different variants of UNet for Gaussian denoising with noise level 25 on Set12 and BSD68 datasets are provided in Table 1. It can be seen that our Converse-UNet achieves better results than Conv-UNet and ConvT-UNet while having the same number of parameters. The possible reason is that our reverse convolution operator can encourage each channel to capture more spatial information. Figure 2 shows visual comparisons between different methods, it can be seen that our Converse-UNet can produce visually pleasant results without introducing artifacts.

B. Application to deblurring task

Apart from the deblurring experiments showcasing the property of the proposed operator in USRNet, we also conduct experiments with Converse-DnCNN to demonstrate its effectiveness as a basic component. In this experiment, blurred images are generated by convolving the clean images with a 7×7 isotropic Gaussian blur kernel with standard deviation 1.6, followed by adding Gaussian noise with a noise level of 2.25. Table 2 reports the average PSNR results of different methods, including DnCNN [14], Conv-DnCNN, ConvT-DnCNN, and Converse-DnCNN, on different datasets. It can be seen that our proposed Converse-DnCNN achieves the best PSNR results. Figure 3 shows the visual comparisons between different methods an example image from Set14. It can be seen that our method can produce visually pleasant results with sharp edges.

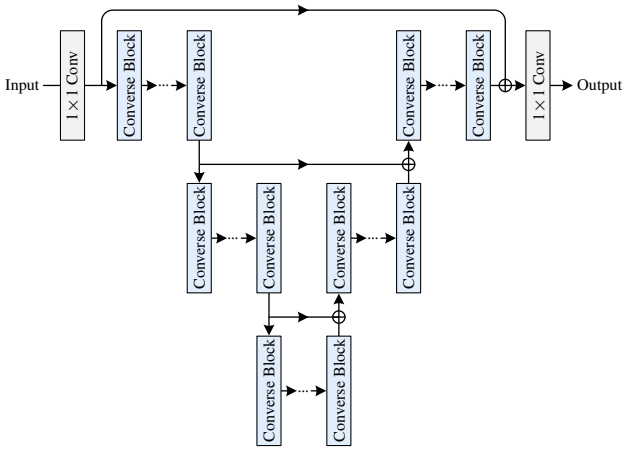


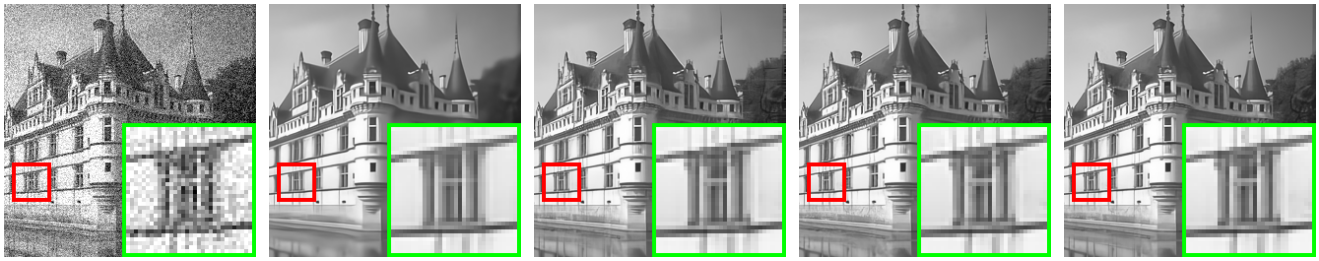
Figure 1. The architecture of the Converse-UNet.

Table 1. The number of parameters and average PSNR(dB) results of different variants of UNet for Gaussian denoising with noise level 25 on Set12 and BSD68 datasets.

Models	# Parameters	Dataset	
		Set12	BSD68
Conv-UNet	734,913	30.62	29.08
ConvT-UNet	734,913	30.62	29.13
Converse-UNet	734,913	30.70	29.37

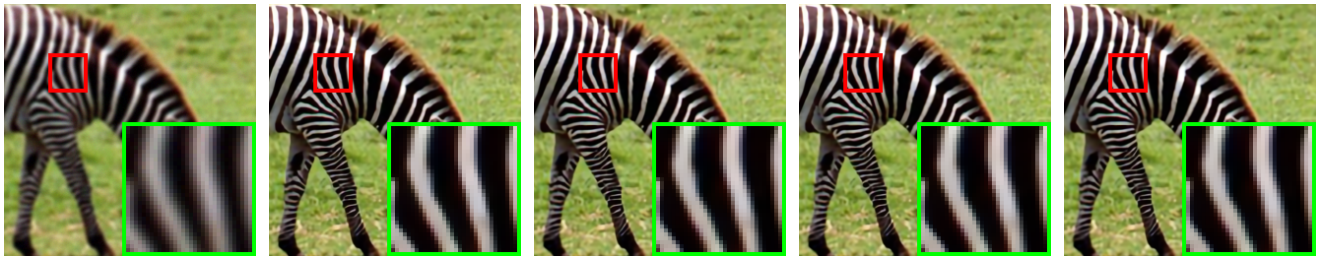
Table 2. The average PSNR(dB) results of different methods for image debluring task.

Models	Dataset		
	Set5	Set14	BSD68
DnCNN [14]	33.09	29.88	29.52
Conv-DnCNN	33.19	29.99	29.57
ConvT-DnCNN	33.08	29.91	29.54
Converse-DnCNN	33.23	30.01	29.61



(a) Noisy image ($\sigma = 25$) (b) DnCNN/30.26dB (c) Conv-UNet/30.05dB (d) ConvT-UNet/30.06dB (e) Converse-Unet/30.33dB

Figure 2. Denoising results of different methods on an example image from BSD68 with noise level 25.



(a) Blurry image (b) DnCNN/30.97dB (c) Conv-DnCNN/31.09dB (d) ConvT-DnCNN/31.05dB (e) Converse-DnCNN/31.25dB

Figure 3. Deblurring results of different methods on the image 'zebra' from Set14.

C. Analysis with a single-layer network

The most straightforward way to demonstrate the effectiveness of the proposed reverse convolution operator is to compare it with the depthwise convolution and depthwise transposed convolution on a single-layer network. To illustrate this, we consider two image restoration tasks: deblurring and super-resolution. In this experiment, blurred images are created by convolving clean images with a 7×7 isotropic Gaussian blur kernel having a standard deviation of 1.6. The low-resolution images are modeled as blurred, downsampled, and noise-free versions of the corresponding high-resolution images. The blurring process remains consistent with the previously described blurring method. The downscaling factor is set to 3. Table 3 provides the average PSNR(dB) results of a single-layer network with different operators for deblurring and super-resolution. We can see that our operator outperforms both depthwise convolution and depthwise transposed convolution for both the deblurring and super-resolution tasks. The reason is that our reverse convolution operator actually models the degradation processes of the deblurring task and super-resolution task.

D. Analysis of computational complexity

The computational complexity of our operator is primarily governed by the FFT and inverse FFT operations, both scaling as $\mathcal{O}(HW \log_2 HW)$, in contrast to the $\mathcal{O}(HWK^2)$ complexity of standard convolution. While standard convolution remains more efficient for small kernel sizes K, our operator demonstrates greater computational efficiency as K increases (see [11] for a detailed discussion on FFT-based acceleration). It is worth noting, however, that standard convolution benefits from highly optimized GPU implementations, whereas our operator currently lacks such low-level optimization, which partially accounts for the observed runtime gap.

Table 3. The average PSNR(dB) results of a single-layer network with different operators for deblurring and super-resolution on set5 dataset.

Models	Tasks	
	Deblurring	Super-Resolution
Transposed	26.90	25.50
Convolution	27.11	25.52
Reverse Convolution	27.38	26.12

E. Datasets and training details

We use widely recognized datasets for image restoration in our experiments. For denoising, we validate our operator on the Set12 [15] and BSD68 (contains 68 images from The Berkeley Segmentation Dataset and Benchmark) [10] datasets. For super-resolution, we use the BSD100 (contains 100 images from The Berkeley Segmentation Dataset and Benchmark) [10], Urban100 [6], Set5 [1], and Set14 [13] datasets. For deblurring, we We follow the training setup from [16] by optimizing network

parameters with L1 loss and the Adam optimizer. The learning rate starts at 1e-4, halves every 100,000 iterations, and stops when below 5e-7.

F. Derivation of equation (2)

Our target optimization problem is as follows:

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \|\mathbf{Y} - (\mathbf{X} \otimes \mathbf{K}) \downarrow_s\|_F^2 + \lambda \|\mathbf{X} - \mathbf{X}_0\|_F^2 , \tag{1}$$

The closed-form solution to Eq. (1) is given below:

$$\mathbf{X}^* = \mathbf{F}^{-1}\left(\frac{1}{\lambda}\left(\mathbf{L} - \overline{\mathbf{F}_K} \odot_s \frac{(\mathbf{F}_K\mathbf{L}) \Downarrow_s}{|\mathbf{F}_K|^2 \Downarrow_s + \lambda}\right)\right) \tag{2}$$

We now present a detailed derivation of Eq. (2).

Assumption 1 The cyclic convolution operator can be represented by a blurring matrix $\mathbf{H}$. It is satisfied provided the underlying blurring kernel is shift-invariant and the boundary conditions make the convolution operator periodic. Using the cyclic convolution assumption, the blurring matrix and its conjugate transpose can be decomposed as

$$\mathbf{H} = \mathbf{F}^{-1}\mathbf{\Lambda}\mathbf{F} \tag{3}$$

$$\mathbf{H}^H = \mathbf{F}^{-1}\mathbf{\Lambda}^H\mathbf{F} \tag{4}$$

where the matrices $\mathbf{F}$ and $\mathbf{F}^{-1}$ are associated with the Fourier and inverse Fourier transforms (satisfying $\mathbf{F}\mathbf{F}^{-1} = \mathbf{F}^{-1}\mathbf{F} = \mathbf{I}_{N_h}$) and $\mathbf{\Lambda} = \text{diag}\{\mathbf{F}_K\} \in \mathbb{C}^{N_h \times N_h}$ is a diagonal matrix, whose diagonal elements are the Fourier coefficients of the first column of the blurring matrix $\mathbf{H}$, denoted as $\mathbf{K}$. Using the decompositions (3) and (4), the blurring operator $\mathbf{H}\mathbf{x}$ and its conjugate $\mathbf{H}^H\mathbf{x}$ can be efficiently computed in the frequency domain.

Assumption 2 The down-sampling operator can be represented by a decimation matrix $\mathbf{S} \in \mathbb{R}^{N_l \times N_h}$, while its conjugate transpose $\mathbf{S}^H \in \mathbb{R}^{N_h \times N_l}$ interpolates the decimated image with zeros. The decimation matrix satisfies the relationship $\mathbf{S}\mathbf{S}^H = \mathbf{I}_{N_l}$. Denoting $\underline{\mathbf{S}} \triangleq \mathbf{S}^H\mathbf{S}$, multiplying an image by $\underline{\mathbf{S}}$ can be achieved by making an entry-wise multiplication with an $N_h \times N_h$ mask having ones at the sampled positions and zeros elsewhere.

Eq. (1) transforms to

$$\mathbf{X}^* = \arg\min_{\mathbf{X}} \|\mathbf{Y} - \mathbf{S}\mathbf{H}\mathbf{x}\|_F^2 + \lambda\|\mathbf{X} - \mathbf{X}_0\|_F^2 \tag{5}$$

whose solution is given by

$$\mathbf{X}^* = (\mathbf{H}^H\underline{\mathbf{S}}\mathbf{H} + 2\lambda)^{-1}(\mathbf{H}^H\mathbf{S}^H\mathbf{Y} + 2\lambda\mathbf{X}_0) \tag{6}$$

with $\underline{\mathbf{S}} = \mathbf{S}^H\mathbf{S}$.

Lemma 1 The following equality holds

$$\mathbf{F}\underline{\mathbf{S}}\mathbf{F}^{-1} = \frac{1}{d}\mathbf{J}_d \otimes \mathbf{I}_{N_l} \tag{7}$$

where $\mathbf{J}_d \in \mathbb{R}^{d\times d}$ is a matrix of ones, $\mathbf{I}_{N_l} \in \mathbb{R}^{N_l \times N_l}$ is the $N_l \times N_l$ identity matrix and $\otimes$ is the Kronecker product.

Using the property of the matrix $\mathbf{F}\underline{\mathbf{S}}\mathbf{F}^H$ given in Lemma 1 and taking into account the assumptions mentioned above, the analytical solution (6) can be rewritten as

$$\mathbf{X}^* = \mathbf{F}^{-1}\left(\frac{1}{d}\underline{\mathbf{\Lambda}}^H\underline{\mathbf{\Lambda}} + 2\lambda\right)^{-1}\mathbf{F}\left(\mathbf{H}^H\mathbf{S}^H\mathbf{Y} + 2\lambda\mathbf{X}_0\right) \tag{8}$$

where the matrix $\underline{\mathbf{\Lambda}} \in \mathbb{C}^{N_l \times N_h}$ is defined as

$$\underline{\mathbf{\Lambda}} = [\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \cdots, \mathbf{\Lambda}_d] \tag{9}$$

and where the blocks $\mathbf{\Lambda}_i \in \mathbb{C}^{N_l \times N_l}$ $(i = 1, \cdots, d)$ satisfy the relationship

$$\text{diag}\{\mathbf{\Lambda}_1, \cdots, \mathbf{\Lambda}_d\} = \mathbf{\Lambda}. \tag{10}$$

Lemma 2 (Woodbury formula) The following equality holds conditional on the existence of $\mathbf{A}_1^{-1}$ and $\mathbf{A}_3^{-1}$

$$\begin{aligned}&(\mathbf{A}_1 + \mathbf{A}_2\mathbf{A}_3\mathbf{A}_4)^{-1}\\ &= \mathbf{A}_1^{-1} - \mathbf{A}_1^{-1}\mathbf{A}_2(\mathbf{A}_3^{-1} + \mathbf{A}_4\mathbf{A}_1^{-1}\mathbf{A}_2)^{-1}\mathbf{A}_4\mathbf{A}_1^{-1}\end{aligned} \tag{11}$$

where $\mathbf{A}_1$, $\mathbf{A}_2$, $\mathbf{A}_3$ and $\mathbf{A}_4$ are matrices of correct sizes.

When Assumption 1 and 2 are satisfied, the solution of Problem (5) can be computed using the following closed-form expression

$$\mathbf{X}^* = \frac{1}{2\lambda}\mathbf{r} - \frac{1}{2\lambda}\mathbf{F}^{-1}\underline{\mathbf{\Lambda}}^H\left(2\lambda d\mathbf{I}_{N_l} + \underline{\mathbf{\Lambda}}\underline{\mathbf{\Lambda}}^H\right)^{-1}\underline{\mathbf{\Lambda}}\mathbf{F}_r \tag{12}$$

where $\mathbf{r} = \mathbf{H}^H\mathbf{S}^H\mathbf{Y} + 2\lambda\mathbf{X}_0$ and $\underline{\mathbf{\Lambda}}$ is defined in (9).

We denote the FFT of $\mathbf{r}$ as $\mathbf{L}$, where $\mathbf{L} = \mathbf{F}_r = \mathbf{\Lambda}^H\mathbf{F}\mathbf{S}^H\mathbf{Y} + 2\lambda\mathbf{F}_{X_0} = \overline{\mathbf{F}_K}\mathbf{F}_{Y\uparrow_s} + \lambda\mathbf{F}_{X_0}$. Additionally, given that $\underline{\mathbf{\Lambda}} = [\mathbf{\Lambda}_1, \mathbf{\Lambda}_2, \ldots, \mathbf{\Lambda}_d] = [\mathbf{F}_{K1}, \mathbf{F}_{K2}, \ldots, \mathbf{F}_{Kd}]$, we deduce that $\underline{\mathbf{\Lambda}}\underline{\mathbf{\Lambda}}^H = \overline{\mathbf{F}_K}\mathbf{F}_K$.

Note that $\mathbf{I}_{N_l} \in \mathbb{R}^{N_l \times N_l}$ is the $N_l \times N_l$ identity matrix. Thus, the down-sampling operation is characterized by $\mathbf{S}\mathbf{I}_{N_l}$, which represents a mask matrix that corresponds to the sampled positions.

Now, we simplify Eq. (12) as follows:

$$
\begin{aligned}
\mathbf{X}^* &= \frac{1}{2\lambda}\mathbf{r} - \frac{1}{2\lambda}\mathbf{F}^{-1}\underline{\mathbf{\Lambda}}^H\left(2\lambda d\mathbf{I}_{N_l} + \underline{\mathbf{\Lambda}\mathbf{\Lambda}}^H\right)^{-1}\underline{\mathbf{\Lambda}}\mathbf{F}_r \\
&= \mathbf{F}^{-1}\left(\frac{1}{2\lambda}\mathbf{L} - \frac{1}{2\lambda}\underline{\mathbf{\Lambda}}^H\left(2\lambda d\mathbf{I}_{N_l} + \overline{\mathbf{F}_K}\mathbf{F}_K\right)^{-1}\underline{\mathbf{\Lambda}}\mathbf{L}\right) \\
&= \mathbf{F}^{-1}\left(\frac{1}{2\lambda}\mathbf{L} - \frac{1}{2\lambda}\underline{\mathbf{\Lambda}}^H\left(2\lambda d\mathbf{I}_{N_l} + \overline{\mathbf{F}_K}\mathbf{F}_K\right)^{-1}\mathbf{F}_r\right) \\
&= \mathbf{F}^{-1}\left(\frac{1}{2\lambda}\mathbf{L} - \frac{1}{2\lambda}\frac{\overline{\mathbf{F}_K}\mathbf{F}_K\mathbf{L}}{2\lambda d\mathbf{I}_{N_l} + |\mathbf{F}_K|^2}\right) \\
&= \mathbf{F}^{-1}\left(\frac{1}{2\lambda}\mathbf{L} - \frac{1}{2\lambda}\overline{\mathbf{F}_K} \odot_s \frac{\mathbf{S}\mathbf{F}_K\mathbf{L}}{2\lambda d\mathbf{S}\mathbf{I}_{N_l} + \mathbf{S}|\mathbf{F}_K|^2}\right) \\
&= \mathbf{F}^{-1}\left(\frac{1}{\lambda}\left(\mathbf{L} - \overline{\mathbf{F}_K} \odot_s \frac{(\mathbf{F}_K\mathbf{L}) \Downarrow_s}{|\mathbf{F}_K|^2 \Downarrow_s + \lambda}\right)\right)
\end{aligned}
\tag{13}
$$

G. More architecture and networks

Considering that our operator performs comparably to convolution and transposed convolution in several image restoration tasks, we aim to further evaluate its capabilities as a fundamental component across a wide range of tasks. Specifically, we focus on two prominent computer vision tasks in high-level vision: classification and segmentation. These tasks require the operator to capture not only pixel-level details but also semantic information. To assess this, we integrate our Converse2D operator into the most well-known architectures, including ResNet [4], DenseNet [5], and FCN [9], replacing the standard convolution layers. Our results demonstrate that the Converse2D operator also performs effectively as a modular component in high-level vision tasks.

G.1. Classification with ResNet architecture

ResNet [4] introduces residual connections to facilitate the training of very deep networks, improving accuracy and convergence by mitigating the vanishing gradient problem. In this experiment, we replace the standard convolution operations in ResNet [4] with our proposed Converse2D operator, resulting in a variant we call Converse-ResNet. We compare it with other two versions: one with depthwise convolution (Conv-ResNet), one with depthwise transposed convolution (ConvT-ResNet). For this experiment, we use a ResNet architecture with 56 layers. Notably, our ResNet variants have significantly fewer parameters due to the use of depthwise convolutions. It can maintain comparable performance by utilizing more channels, thereby keeping the number of parameters roughly the same. The average accuracies on CIFAR-10 and CIFAR-100 datasets are shown in Table 4. It can be observed that our operator achieves performance comparable to the other two, demonstrating its ability to effectively capture semantic information.

Table 4. The average accuracies of different variants of ResNet for classification on CIFAR-10 and CIFAR-100 datasets.

Models	Dataset	
	CIFAR-10 [7]	CIFAR-100 [7]
Conv-ResNet	82.62	63.01
ConvT-ResNet	81.65	63.14
Converse-ResNet	83.89	63.12

G.2. Classification with DenseNet architecture

The key innovation of DenseNet [5] is dense connectivity pattern, where each layer directly receives inputs from all preceding layers. In this experiment, we replace the standard convolution operations in each DenseLayer with our proposed Converse2D operator, referring to the modified network as Converse-DenseNet. We compare two variants incorporating depthwise convolution and depthwise transposed convolution, which we refer to as Conv-DenseNet and ConvT-DenseNet, respectively. The average accuracies on CIFAR-10 and CIFAR-100 datasets are shown in Table 5. The results demonstrate the effectiveness of our proposed operator. We believe that dense connectivity may further benefit from our operator, as it can capture more single-channel information and aggregate it through the 1×1 convolution.

Table 5. The average accuracies of different variants of DenseNet for classification on CIFAR-10 and CIFAR-100 datasets.

Models	Dataset	
	CIFAR-10 [7]	CIFAR-100 [7]
Conv-DenseNet	91.08	75.38
ConvT-DenseNet	91.12	75.22
Converse-DenseNet	91.16	75.43

G.3. Segmentation with FCN architecture

FCN [9] replaces the fully connected layers in traditional CNNs with convolutional layers and employs upsampling to restore the image size lost due to convolution and pooling. In this experiment, we substitute the bilinear interpolation in FCN-8s with our proposed operator to achieve the upsampling process, which we refer to as Converse-FCN-8s. In terms of the convolution variant (Conv-FCN-8s), we use 3×3 depthwise convolutions and nearest-neighbor interpolation to perform the upsampling operation. The transposed convolution variant (ConvT-FCN-8s) performs upsampling via depthwise transposed convolutions, replacing the bilinear interpolation. We choose the widely used PASCAL VOC [3] and CamVid [2] datasets to train and test the model. The average pixel accuracies are presented in Ta-

ble 6. The results are promising, as image segmentation requires capturing both pixel-level and image-level information. This demonstrates that our operator can serve as a fundamental component across a wide range of tasks.

Table 6. The average pixel accuracies of different variants of FCN-8s for Segmentation on PASCAL VOC and CamVid datasets.

Models	Dataset	
	PASCAL VOC [3]	CamVid [2]
Conv-FCN-8s	84.6	87.5
ConvT-FCN-8s	84.5	87.3
Converse-FCN-8s	84.8	87.7

G.4. Depth estimation with BTS architecture

The key component of BTS [8] is the use of local planar guidance layers combined with a backbone encoder-decoder architecture, enabling precise depth estimation from a single RGB image. In this experiment, we adopt BTS as our baseline model and replace its standard convolution operations with our proposed Converse2D operator, resulting in a modified version referred to as Converse-BTS. We evaluate two variants using depthwise convolution and depthwise transposed convolution in the decoder stages, named Conv-BTS and ConvT-BTS, respectively. The evaluation is conducted on the NYU Depth v2 dataset, and the quantitative results are summarized in Table 7. Our results indicate that the proposed operator outperforms both the depthwise convolution and depthwise transposed convolution variants. We attribute these improvements to the enhanced capacity of Converse2D to preserve spatial structure and promote better single-channel feature learning, which is crucial in dense prediction tasks such as monocular depth estimation.

Models	higher is better	lower is better		
	$\delta < 1.25$	Abs Rel	RMSE	Sq Rel
Conv-BTS	0.880	0.109	0.391	0.063
ConvT-BTS	0.879	0.111	0.390	0.061
Converse-BTS	**0.887**	**0.105**	**0.385**	**0.059**

Table 7. Evaluation results of different BTS variants for **depth estimation** on the NYU Depth v2 dataset.

H. More Visualization

We visualize the kernel patterns by plotting the learned kernels in the 10-th Converse2D module of Converse-DnCNN, as shown in Figure 4. It can be seen that the kernel patterns exhibit substantial diversity, which enables the network to capture richer semantic information and contributes to the module's efficiency.

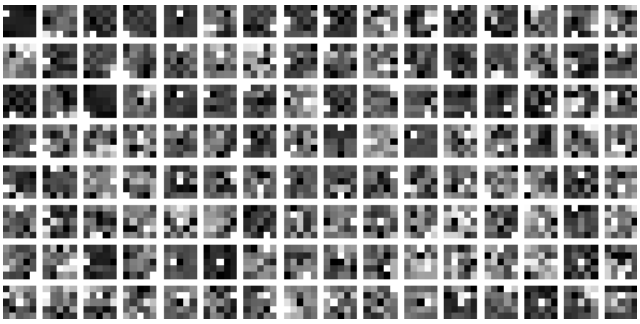
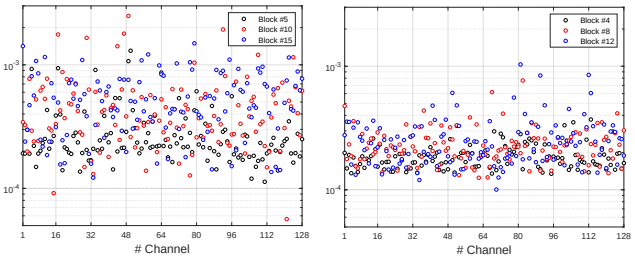


Figure 4. The kernel visualization of the 10-th Converse2D operator in Converse-DnCNN.

Figure 5 presents the learned λ values across different Converse2D operators in Converse-DnCNN and Converse-SRResNet. Here are two key observations. First, λ mainly ranges from 1×10^{-4} to 1×10^{-3}. Second, the learned λ of Converse-DnCNN differs from those of Converse-SRResNet. This divergence likely originates from the denoising-specific requirements, where noise in the feature map necessitates a larger λ to mitigate the ill-poseness of Eq. (1).



(a) (b)

Figure 5. The visualization of the learned λ in different Converse2D operators for (a) Converse-DnCNN and (b) Converse-SRResNet.

(a) ConvNet/20.49dB (b) ConverseNet/22.73dB

Figure 6. Deblurring results of different methods on the image "building" from Urban100 with noise level 2.55.

References

[1] Marco Bevilacqua, Aline Roumy, Christine Guillemot, and Marie Line Alberi-Morel. *Low-complexity single-image*

(a) ConvNet/20.75dB (b) ConverseNet/20.21dB

Figure 7. Deblurring results of different methods on the image "building" from Urban100 with noise level 2.55.

(a) ConvNet/17.68dB (b) ConverseNet/20.36dB

Figure 8. Deblurring results of different methods on the image "building" from Urban100 with noise level 2.55.

(a) ConvNet/27.26dB (b) ConverseNet/29.75dB

Figure 9. Deblurring results of different methods on the image "building" from Urban100 with noise level 2.55.

super-resolution based on nonnegative neighbor embedding. BMVA press, 2012. 2

[2] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. Semantic object classes in video: A high-definition ground truth database. *PRL*, 30(2):88–97, 2009. 4, 5

[3] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88:303–338, 2010. 4, 5

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[5] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 4

[6] Jia-Bin Huang, Abhishek Singh, and Narendra Ahuja. Single image super-resolution from transformed self-exemplars. In *CVPR*, pages 5197–5206, 2015. 2

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[8] Jin Han Lee, Myung-Kyu Han, Dong Wook Ko, and Il Hong Suh. From big to small: Multi-scale local planar guidance for monocular depth estimation. *arXiv preprint arXiv:1907.10326*, 2019. 5

[9] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 4

[10] David Martin, Charless Fowlkes, Doron Tal, and Jitendra Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *ICCV*, pages 416–423, 2001. 2

[11] Michaël Mathieu, Mikael Henaff, and Yann LeCun. Fast training of convolutional networks through ffts. In *ICLR*, 2014. 2

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 1

[13] Roman Zeyde, Michael Elad, and Matan Protter. On single image scale-up using sparse-representations. In *ICCS*, pages 711–730, 2010. 2

[14] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE TIP*, 26(7):3142–3155, 2017. 1

[15] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE TIP*, 26(7):3142–3155, 2017. 2

[16] Kai Zhang, Yawei Li, Wangmeng Zuo, Lei Zhang, Luc Van Gool, and Radu Timofte. Plug-and-play image restoration with deep denoiser prior. *IEEE TPAMI*, 44(10):6360–6376, 2021. 2