

# Wide kernels and their DCT compression in convolutional networks for nuclei segmentation<sup>☆</sup>

Vincent Andrearczyk<sup>\*,1</sup>, Valentin Oreiller, Adrien Depeursinge<sup>2</sup>

University of Applied Sciences of Western Switzerland HES-SO Valais, Rue de Technopole 3, 3960 Sierre, Switzerland  
Service of Nuclear Medicine and Molecular Imaging, CHUV, Lausanne, Switzerland

## ARTICLE INFO

### Keywords:

Convolutional network  
Segmentation  
Receptive field  
Discrete cosine transform

## ABSTRACT

The locality and spatial field of view of image operators have played a major role in image analysis, from hand-crafted to deep learning methods. In Convolutional Neural Networks (CNNs), the field of view is traditionally set to very small values (e.g.  $3 \times 3$  pixels) for individual kernels and grown throughout the network by cascading layers. Automatically learning or adapting the best spatial support of the kernels can be done by using large kernels. Due to the computation requirements of standard CNN architectures, this has been little investigated in the literature. However, if large receptive fields are needed to capture wider contextual information on a given task, it could be learned from the data. Obtaining an optimal receptive field with few layers is very relevant in applications with a limited amount of annotated training data, e.g. in medical imaging.

We show that CNNs (2D U-Nets) with large kernels outperform similar models with standard small kernels on the task of nuclei segmentation in histopathology images. We observe that the large kernels mostly capture low-frequency information, which motivates the need for large kernels and their efficient compression via the Discrete Cosine Transform (DCT). Following this idea, we develop a U-Net model with wide and compressed DCT kernels that leads to similar performance and trends to the standard U-Net, with reduced complexity. Visualizations of the kernels in the spatial and frequency domains, as well as the effective receptive fields, provide insights into the models' behaviors and the learned features.

## 1. Introduction

Descriptors localized in space, i.e. defined by a limited spatial support, have been extensively used in computer vision [1]: e.g. filter banks [2], wavelets [3], Local Binary Patterns (LBP) [4]. In Convolutional Neural Networks (CNN), the receptive fields (equivalent to the spatial support) of the neurons grow throughout the network and are manually designed (i.e. hardcoded) by kernel sizes and pooling operations. Small kernels (e.g.  $3 \times 3$ ) are commonly used in deep learning [5,6] due to the quick increase (quadratic or cubic growth in 2D/3D, respectively) of trainable parameters as well as number of operations for computing the convolution when the size increases. Recent literature suggested that large kernels may, however, be important and that cascading small kernels do not empower the network to learn all types of discriminative features [7,8]. In [9], the authors show that a large kernel design requires fewer layers to obtain large Effective

Receptive Fields (ERF). An extension to 3D kernels, in which depth-wise convolutions are ineffective, is proposed in [10] with spatial-wise partition convolutions. Other relevant works on the importance of kernel size include the application to super-resolution [11], the proposition of a trainable kernel size in [12], deformable convolutions [13] and deformable kernels [14].

Recently, kernels with parametric representations have been used in deep learning, e.g. Discrete Cosine Transform (DCT) [15–18], Circular Harmonics (CH) [19,20] in 2D, and Spherical Harmonics (SH) [21, 22] in 3D. These architectures can provide, among others, a built-in equivariance to input transformations and a reduction of trainable parameters.

More recently, Vision Transformers (ViT), with long-term spatial dependencies, have obtained state-of-the-art results in various tasks in computer vision. Based on ViT, the Global Filter Network (GFNet) proposed in [23] learns long-range spatial dependencies in the frequency

<sup>☆</sup> This work was funded by the Hasler Foundation (project No 21064) and the Swiss National Science Foundation (SNSF, grant 205320\_179069).

<sup>\*</sup> Corresponding author at: University of Applied Sciences of Western Switzerland HES-SO Valais, Rue de Technopole 3, 3960 Sierre, Switzerland.  
E-mail address: [vincent.andrearczyk@hevs.ch](mailto:vincent.andrearczyk@hevs.ch) (V. Andrearczyk).

<sup>1</sup> Researcher.

<sup>2</sup> Co-ordinator.

domain with log-linear complexity. CNNs are not designed to capture long-range dependencies between different image regions. Although very long-range dependencies may not be required for all imaging tasks, an optimal locality can be beneficial. For instance, nuclei segmentation may require a receptive field of approximately the nucleus size. Besides, CNNs tend to outperform ViTs in low data regimes (unless intensive pre-training is used) [24].

The use of DCT compression of large CNN kernels has the potential to reduce the computational complexity and speed up model convergence, yet has not been investigated in the literature. In this paper, we investigate the value and influence of large kernels in CNNs and their compression by DCT for the segmentation of multi-organ nuclei in histopathology images. This task is particularly suited for the evaluation of the proposed models for two reasons. (i) We expect a medium-sized optimal receptive field, up to the size of the nucleus; and (b) the limited amount of data requires fast convergence and low computational complexity. CNNs are computationally expensive, require a lot of training data, and are complex to interpret (often referred to as “black boxes”) due to the repeated cascading operations including convolutions and pooling operations. The large parametric kernels with DCT compression with a small number of layers help to reduce the complexity, fasten the convergence, and better understand the inner mechanism of the models.

The structure of the paper is as follows. The proposed architecture and DCT-compressed wide kernels as well as the dataset, methods and metrics for nuclei segmentation in histopathological images are introduced in Section 2. The results obtained with varying kernel sizes and the impact of DCT convolutions are presented in Section 3, together with spectral and spatial visualizations of the learned kernels. Finally, discussions and a conclusion are proposed in Section 4.

## 2. Methods

We first introduce the general segmentation architecture, training and postprocessing in Section 2.1. DCT-compressed kernels and the corresponding proposed convolutional layer are detailed in Sections 2.2 and 2.3, respectively. The dataset of histopathological images with the annotated nuclei is detailed in Section 2.4 and evaluation metrics are explained in Section 2.5.

### 2.1. Network architecture and training

Following [25], we use a light-weight U-Net [26] which contains two down-sampling levels with standard shortcut connections between the encoder and decoder layers. The architecture is illustrated in Fig. 1. The encoder path includes a first stem convolution as suggested in [7], followed by two convolutional layers with  $2 \times 2$  max-pooling to reduce the spatial dimension and another convolution at the bottleneck. The numbers of feature maps for these layers are 8 (stem), 32, 64 and 64. The decoder path contains two convolutional layers, both preceded by a  $2 \times 2$  bi-linear upsampling. The convolutional layers use a kernel size of  $5 \times 5$ , except for the layer after the stem for which we vary the kernel size from  $5 \times 5$  to  $21 \times 21$ . We use wide kernels only in the first layer to quickly achieve large receptive fields in the first level of the U-Net. The convolutional layers are also connected to a batch normalization layer followed by a ReLU activation. The output segmentation prediction is encoded in the final layer by a  $1 \times 1$  convolution with softmax activation. The prediction is modeled as a three classes probability, namely nucleus core, nucleus border, and background.

The output of the U-Net is post-processed to obtain an instance segmentation of each nucleus. This post-processing consists in taking the maximum of the three prediction probabilities, then using the core and border predictions as seed and landscape respectively for a

watershed algorithm<sup>3</sup> [27]. The borders are thus only used for this watershed algorithm and the post-processed predictions are binary, namely nucleus and background. The evaluation metrics are computed on these binary post-processed predictions. An example of a pre- and post-processed prediction is illustrated in Fig. 2.

The U-Net models are optimized with an Adam optimizer to minimize the class-balanced cross-entropy at an initial learning rate of  $10^{-3}$ . The models are trained on patches randomly cropped from the training set with a batch size of 16. Without padding, convolving with large kernels results in smaller output maps which limits the computation of the loss. To maintain the output size constant for different kernel sizes, we vary the size of the patches from  $64 \times 64$  pixels for the smallest  $5 \times 5$  kernels to  $80 \times 80$  for the largest  $21 \times 21$  pixels. This operation is equivalent to padding the images for the convolution operations with values taken from neighboring regions of the crop, rather than using zero-padding or other methods. In addition to this random cropping, data augmentation is performed by random  $90^\circ$  rotation and random brightness shift. The training is conducted for a maximum of 200 epochs, with early stopping based on the validation  $F_1$ -score (see Section 2.5). The experiments are performed on a Nvidia V100. The code for the implementation is available on our GitHub.<sup>4</sup>

### 2.2. DCT-compressed kernels

We consider 2D images as functions  $I \in L_2(\mathbb{R}^2)$ , where each  $I(\mathbf{x}) \in \mathbb{R}$  represents the pixel intensity at a location  $\mathbf{x} = (x_1, x_2) \in \mathbb{R}^2$ .

A DCT kernel  $f_{\text{DCT}} \in \mathbb{R}^{k \times k}$  is defined in the discretized spatial coordinates  $\mathbf{x} = [x_1, x_2]$  for a maximal degree  $N$  as

$$f_{\text{DCT}}(\mathbf{x}) = \sum_{n_1=0}^{N-1} \sum_{n_2=0}^{N-1} w_{n_1, n_2} \psi_{n_1, n_2}(\mathbf{x}), \quad (1)$$

with coefficients  $w_{n_1, n_2} \in \mathbb{R}$ , frequencies  $n_1, n_2$ . The DCT functions  $\psi_{n_1, n_2} \in \mathbb{R}^{k \times k}$  are  $L_1$ -normalized.

$$\psi_{n_1, n_2} = \frac{\phi_{n_1, n_2}}{\|\phi_{n_1, n_2}\|}, \quad (2)$$

where  $\phi_{n_1, n_2}$  are the orthogonal basis functions defined as

$$\phi_{n_1, n_2}(x_1, x_2) = \cos \left[ \frac{\pi}{N} \left( n_1 + \frac{1}{2} \right) x_1 \right] \cos \left[ \frac{\pi}{N} \left( n_2 + \frac{1}{2} \right) x_2 \right]. \quad (3)$$

Using DCT compressed kernels offers several advantages. Firstly, it significantly reduces the number of parameters needed to define a kernel compared to standard 2D kernels. In particular, it provides an orthogonal low-pass approximations of non-parametric CNN kernels. This reduction of parameters is specifically obtained by truncation of the expansion [17]. This makes it a valuable tool that has long been used for image compression. Additionally, the linearity of DCT compressed kernels in Eq. (1) can be efficiently exploited for the convolutions. Once the input is convolved with the family of DCT functions  $\psi_{n_1, n_2}$ , any kernel can be reconstructed (i.e. learned) using a linear recombination of their response maps. This obviates the need to re-convolve with large kernels, as shown in Eq. (4).

These two properties constitute important advantages for learning wide kernels and efficiently computing their responses to a given input.

### 2.3. DCT convolutional layer

After evaluating a standard U-Net with varying kernel sizes (as described in the previous section), we replace the first convolutional layer with a DCT convolutional layer. The latter is implemented by convolving the input feature maps with the family of DCT functions

<sup>3</sup> [docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.watershedIFT.html](https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.watershedIFT.html), as of March 2023.

<sup>4</sup> [https://github.com/vandrearczyk/2d\\_wide\\_dct\\_cnn](https://github.com/vandrearczyk/2d_wide_dct_cnn), Oct. 2023.

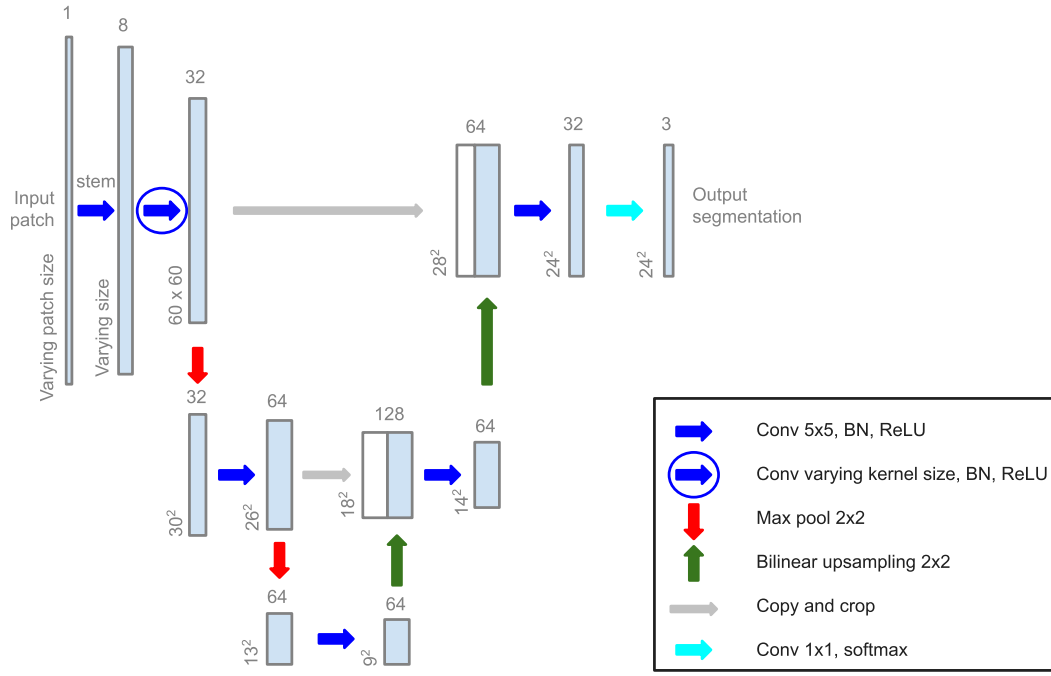


Fig. 1. The architecture of the light-weight U-Net with varying kernel sizes. The kernel size is varied in the second convolutional layer after the stem layer. The size of the input patch is varied based on the kernel size to obtain a constant  $60 \times 60$  size after this second layer without artificial padding.

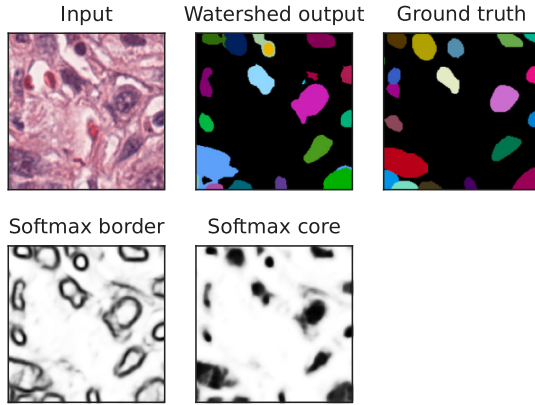


Fig. 2. Example of a wide-kernel ( $k = 21$ ) U-Net prediction. The  $F_1$ -score on this patch is 0.75. Individual colors are used in the top row for each nucleus instance. The U-Net softmax predictions are illustrated in the second row. The predictions post-processed with the watershed algorithm are illustrated on the top row between the input and the ground truth images. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(Eqs. (2) and (3)). These basis kernels are illustrated in Fig. 3. We fix the maximum frequency  $N \in [0, k]$ , with  $k$  the kernel size of the convolution kernel. The inputs are thus convolved with the  $N^2$  basis functions. The response maps ( $I * \psi_{n_1, n_2}$ ) are then recombined using trainable coefficients  $w_{n_1, n_2, i}$  (see Section 2.2) for each output feature map (i.e. channel)  $h_i(\mathbf{x})$ . This process of convolution followed by recombination is equivalent to convolving the inputs with kernels  $f_{DCT, i}$  as defined in Eq. (1). We have

$$\begin{aligned}
 h_i(\mathbf{x}) &= (I * f_{DCT, i})(\mathbf{x}) \\
 &= \sum_{n_1, n_2} w_{n_1, n_2, i} (I * \psi_{n_1, n_2})(\mathbf{x}) \\
 &= I(\mathbf{x}) * \sum_{n_1, n_2} w_{n_1, n_2, i} \psi_{n_1, n_2}(\mathbf{x}),
 \end{aligned} \tag{4}$$

where the response maps of the DCT kernels are computed only once. It is worth noting that the models can be trained with sparsity constraints (e.g.  $L_1$  regularization) on these coefficients. However, in our experiments, we did not observe significant benefits from using this sparsity constraint. Nevertheless, it is an alternative approach to simply limiting the maximal frequency as described next.

Reducing computational complexity can be achieved by selecting subsets of DCT kernels to approximate the 2D kernels. A first approach is to simply restrict the number of maximum frequency  $N$ , yielding low-pass approximations of the kernels. Alternatively, a subset of basis kernels  $\psi_{n_1, n_2}$ , such that  $n_1 + n_2 < N$ , can be used. This reduces the complexity by compressing the kernels, discarding frequency with known marginal contribution as suggested in [18]. This technique is referred to as  $\lambda$ -truncation. The number of required basis functions reduces from  $N^2$  to  $\frac{N(N+1)}{2}$ , as illustrated by the coefficients represented in Figs. 6 (h) and (k), and in [18] (Fig. 2).

#### 2.4. Dataset

For the experiments, we employ a subset of the MoNuSeg 2018 dataset [28] which consists of 24 Hematoxylin and Eosin (H&E) stained images. The images are selected from whole slide images acquired at the 40 $\times$  magnification provided by The Cancer Genome Atlas [29]. This subset contains six  $1000 \times 1000$  images for each of the four tissue types: breast, liver, kidney and prostate. Nuclei instance annotations are provided for these 24 images. We split the data as in [20,30], with  $4 \times 3$  images for training,  $4 \times 1$  for validation and  $4 \times 2$  for test. We repeat the experiments with ten random splits to evaluate the variation of model performance. The H&E images are normalized using the unsupervised stain separation method based on singular value decomposition to learn stain vectors in the optical density color space, as described in [31]. We adapted the code from<sup>5</sup> to fit the needs of our data and experiments. An example of a pre-processed input patch is illustrated in Fig. 2.

<sup>5</sup> [https://github.com/schaugf/HEnorm\\_python](https://github.com/schaugf/HEnorm_python), as of March 2023.

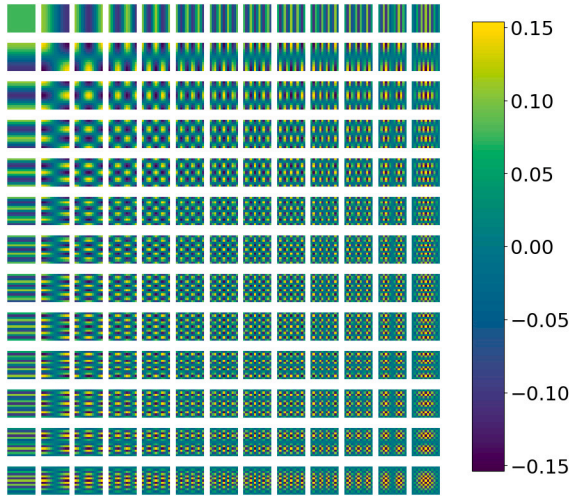


Fig. 3. Examples of  $13 \times 13$  basis kernels  $\psi_{n_1, n_2}$  used in the proposed DCT layer with a maximum frequency  $N = 13$ .

## 2.5. Metrics and evaluation

We use the  $F_1$ -score to evaluate model performance.

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

A predicted nucleus that overlaps (intersection over union) for more than 50% with a ground-truth nucleus is considered as a match to calculate the precision and recall. All experiments are performed on the same ten train/validation/testing splits, described in Section 2.4, to assess and compare performance variation over repetitions across models.

## 2.6. Statistical analyses

For all results, bootstrapping is performed on the 10 splits predictions with 1000 repetitions. The 95% confidence intervals (CI) are reported.

## 3. Results

We first focus on the impact of kernel size on segmentation performance in Section 3.1. Section 3.2 then investigates the relevance of DCT compression of the kernels. In the two steps, we present visualizations in both spatial and frequency domains for interpreting the relevance and coverage of the spatial spectrum of the kernels.

### 3.1. Large kernels

#### 3.1.1. Effect of kernel size

In the first experiment, we evaluate the effect on segmentation performance of increasing the kernel size in the U-Net described in Fig. 1. The nuclei segmentation results for kernel sizes  $k$  ranging from 5 to 21 are reported in Table 1. Increasing the kernel size improves the model performance up to  $k = 17$ , resulting in an optimal  $F_1$ -score of  $0.7312 \pm 0.0254$ , outperforming smaller kernels (e.g.  $k = 5$  with  $F_1$ -score of  $0.7089 \pm 0.0362$ ).

Unlike suggested in [7], depth-wise convolutions and residual connections did not improve the performance when using large kernels and are not reported in this paper.

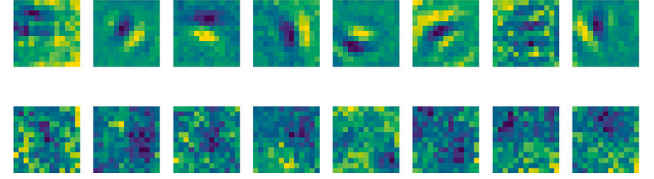
#### 3.1.2. Spatial and frequency analysis of the kernels

The learned kernels and their DCT transform are illustrated in Fig. 4. For these visualizations, we employ a smaller model to enable the

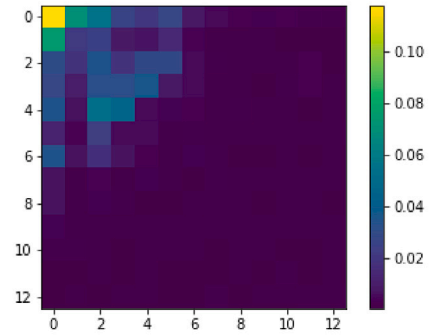
Table 1

Results of test  $F_1$ -scores for varying kernel sizes  $k$ . The mean of the 10 train/test split repetitions is reported alongside 95% CIs. The number of trainable parameters in the wide kernels layer (first layer after the stem layer) is also reported.

$k$	$F_1$ -score	# train. param.
5	0.7089 [0.6844, 0.7276]	424
9	0.7109 [0.6956, 0.7243]	1336
13	0.7285 [0.7122, 0.7427]	2728
17	0.7312 [0.7154, 0.7457]	4648
21	0.7249 [0.7098, 0.7398]	7080



(a) Trained kernels



(b) Average DCT power spectrum

Fig. 4. Trained 2D kernels (a) and their averaged DCT transform (b) in a standard U-Net described in Section 2.1 with a kernel size of 13. The average of the  $2 \times 8 = 16$  sets of squared coefficients is plotted here for one trained model. Similar coefficients are obtained for other splits.

visualization of all kernels in the wide kernel layer (with 2 input channels and 8 output channels). Similar behavior is obtained when visualizing the model with all channels. Most of the power spectrum is contained in low frequencies, for  $N \leq 7$ , and particularly  $n_1 + n_2 \leq 7$ , the coefficients in the top-left triangle.

These visualizations justify the use of large kernels needed to encode this low-frequency signal. It also motivates the use of DCT compression. Indeed, it appears that too many parameters are used in such wide kernels, and they can be compressed by discarding high frequencies. Finally, the very low coefficients for  $n_1 + n_2 > 7$  also motivate the  $\lambda$ -truncation as proposed in [18].

#### 3.1.3. ERF visualization

Examples of ERFs [9] before and after training a standard U-Net with different kernel sizes are illustrated in Fig. 5. For both small and large kernels, the ERF of the model grows during training, and it is much larger when using large kernels of  $21 \times 21$ . These visualizations show that the ERFs grow throughout the training, as observed in [9], and that larger kernels enable to start the training at an ERF that seems closer to an optimal size, and to grow larger with training.

## 3.2. DCT kernels

### 3.2.1. Kernel size

We present the results of the light-weight U-Net illustrated in Fig. 1, in which we replace the convolution with varying kernel sizes by a DCT



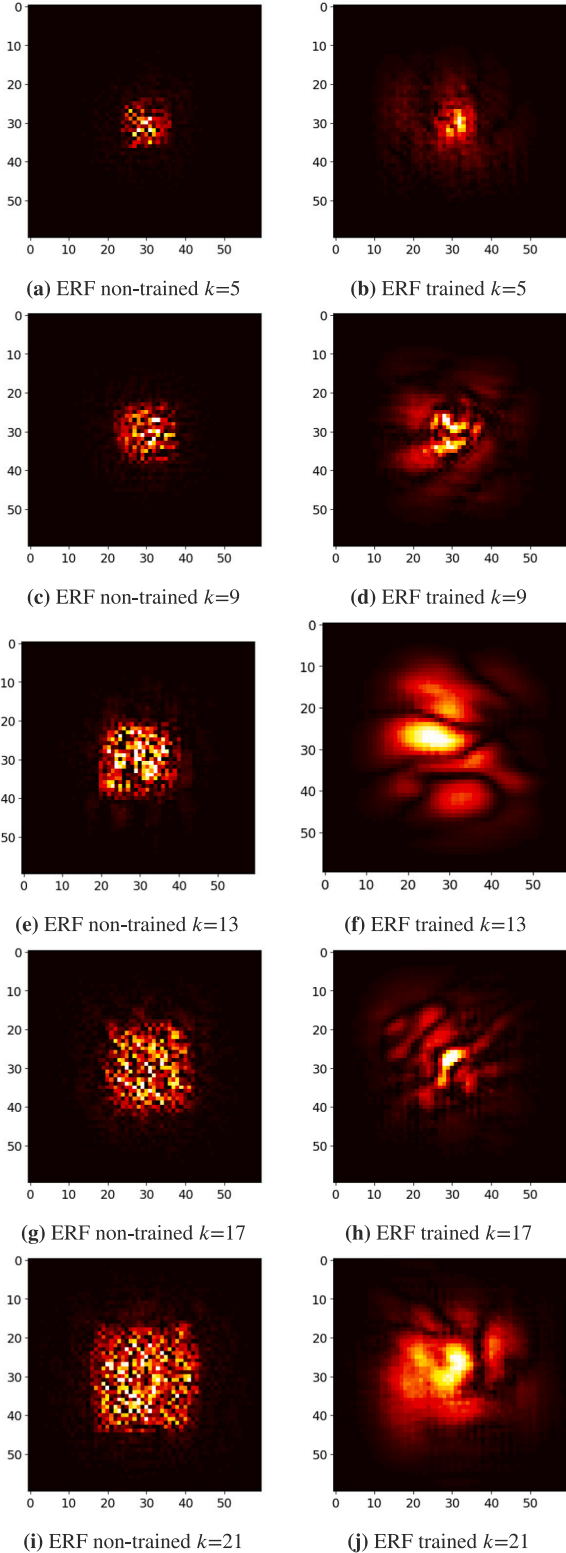


Fig. 5. Visualizations of ERFs of randomly initialized and trained networks with various kernel sizes in the second convolutional layer.

convolution as described in Section 2.3. The results for varying kernel sizes using the DCT kernels described in Section 2.3 are reported in

Table 2

Results of test  $F_1$ -scores for varying kernel sizes  $k$  in a DCT U-Net. A maximum of  $N = 5$  frequencies is used, with  $\lambda$ -truncation. The mean of the 10 train/test split repetitions is reported alongside 95% CIs. The number of trainable parameters (constant) in the DCT layer is also reported.

$k$	$N$	$F_1$ -score	# train. param.
5	5	0.7135 [0.6919, 0.7322]	264
9	5	0.7120 [0.6893, 0.7326]	264
13	5	0.7183 [0.7047, 0.7304]	264
17	5	0.7230 [0.7079, 0.7390]	264
21	5	0.7166 [0.7039, 0.7291]	264

Table 2 together with the number of parameters and convolutions in the DCT layer. The best  $F_1$ -score ( $0.7230 \pm 0.0272$ ) is obtained with  $k = 17$ , similar to the standard convolution as shown in Table 1. The number of trainable parameters is largely reduced as compared to the U-Net with standard wide kernels (fixed with 264 trainable parameters regardless of the kernel size, vs. up to 7080 parameters for the standard convolution with  $k = 21$ ).

### 3.2.2. DCT coefficients

The learned coefficients of a DCT layer are illustrated in Fig. 6. Similarly to the visualizations of non-DCT kernels in Section 3.1.2, we employ a small U-Net to enable the representation of all kernels in the wide kernel layer.

### 3.2.3. Reconstruction of the learned kernels

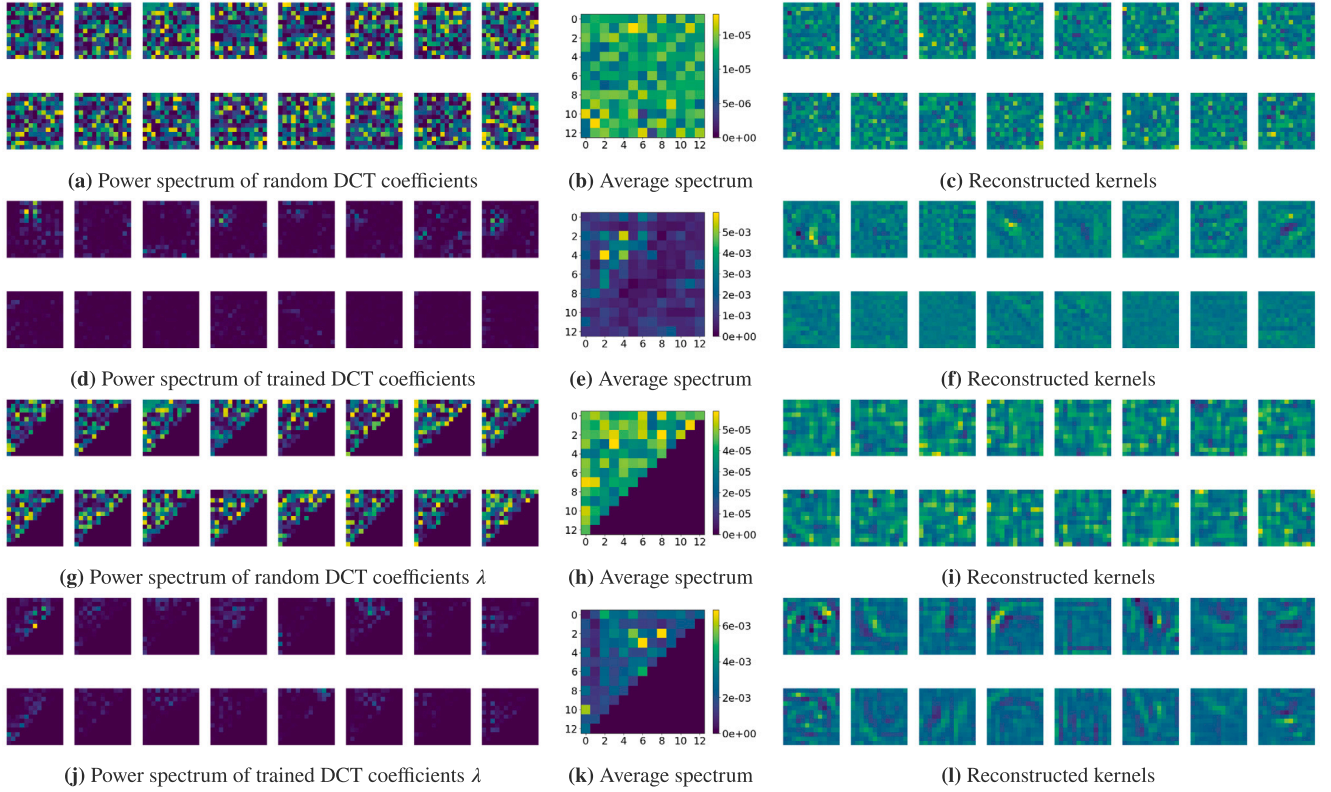
The reconstructed kernels in the spatial domain are illustrated in Fig. 6. The reconstructed kernels are comparable with the kernels of standard, non-DCT kernels illustrated in Fig. 4. The kernels in Fig. 6(l) seem to capture large regions of nuclei, including borders and different textures.

## 4. Discussions and conclusion

We investigated the importance of kernel size and observed that large kernels (up to  $17 \times 17$ ) improve performance over small kernels ( $5 \times 5$ ) in the task of nuclei segmentation in histopathology images with U-Net, reaching an average  $F_1$ -score of 0.7312 (see Table 1). However, the difference is not statistically significant where the 95% CIs are slightly overlapping. This segmentation task requires a relatively constant receptive field, sufficiently extended in the locality of the nucleus, but without the need for very long-range dependencies. Various other tasks may benefit from such investigation of the kernel size and receptive field, for which our work could provide directions.

Following up on the analysis of kernel sizes, we observed that the DCT coefficients of the trained large kernels are concentrated in the low frequencies (see Fig. 4), motivating the use of (large) DCT kernels with built-in compression of frequencies. We showed that similar performance is obtained when DCT kernels with few frequencies are used instead of standard 2D kernels (see Table 2). The best results are also obtained with a kernel size of  $17 \times 17$ .

The DCT compression can largely reduce the model complexity. The number of trainable parameters is reported in Tables 1 and 2. This number can be extremely low when using DCT layers (264 parameters regardless of the kernel size in this experiment), as compared to the standard convolutional layer (up to 7080 parameters with  $k = 21$  for a similar layer). This is explained by the fact that the model only learns scalar weights used for the linear recombinations of feature maps obtained by convolving the input with the basis functions in Eq. (4). As mentioned in Section 2, the number of convolutions is  $(c_{in} \cdot c_{out})$  for standard convolutional layers (i.e. the number of kernels),  $(c_{in} \cdot N^2)$  for the DCT layer (i.e. number of basis DCT functions for a chosen number of frequencies  $N$ ), and  $(N(N+1)/2 \cdot c_{in})$  for a DCT layer with a truncated number of basis DCT functions with  $\lambda = N$ . Besides the comparison of trainable parameters, we report and compare



**Fig. 6.** DCT coefficients (left) and their corresponding average power spectrum (center) and reconstructed kernels (right) for different models. The Figure is organized by rows: (a, b, c) random model; (d, e, f) trained model; (g, h, i) random model with  $\lambda$ -truncation; and (j, k, l) trained model with  $\lambda$ -truncation.

the number of convolutions required for different settings of standard convolutional layers and DCT layers in [Appendix](#). In this context, the DCT layer is beneficial when the number of output feature maps is large and a small number of frequencies is used. It can become impractical, however, for large numbers of frequencies and input feature maps.

Finally, with 3D images and 3D models, DCT kernels could even more drastically reduce the number of trainable parameters (few scalar coefficients vs. large number of voxels constituting wide 3D kernels). However, the number of convolutions with basis kernels and, in particular, the memory used by the responses to these convolutions would be the bottleneck unless strongly compressed by cutting the number of frequencies.

In future work, it will be interesting to combine the wide kernel and DCT compression with rotation equivariance properties [\[20\]](#).

#### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

#### Acknowledgments

This work was supported by the Hasler Foundation (project number 21064) and the Swiss National Science Foundation (SNSF, grant 205320\_179069).

#### Appendix. Comparison of number of convolutions

In [Table 3](#), we report the number of convolutions for different scenarios for the same number of input and output feature maps  $c_{in} = c_{out}$ . These numbers are regardless of the kernel size  $k$ , besides the fact

**Table 3**

Comparison of number of convolutions between a standard convolutional layer ( $c_{in}c_{out}$ ) and a DCT layer ( $c_{in}N^2$ ) with a varying number of maximum frequency  $N$ . The number of input and output channels is equal and varies from 4 to 64.

$c_{in} = c_{out} =$	4	8	16	32	64
Standard conv.					
	16	64	256	1024	4096
DCT conv.					
N					
5	100	200	400	800	1600
9	324	648	1296	2592	5184
13	676	1352	2704	5408	10816
17	1156	2312	4624	9248	18496
21	1764	3528	7056	14112	28224
25	2500	5000	10000	20000	40000
29	3364	6728	13456	26912	53824
33	4356	8712	17424	34848	69696
DCT conv. $\lambda$ -truncated					
$\lambda$					
5	60	120	240	480	960
9	180	360	720	1440	2880
13	364	728	1456	2912	5824
17	612	1224	2448	4896	9792
21	924	1848	3696	7392	14784
25	1300	2600	5200	10400	20800
29	1740	3480	6960	13920	27840
33	2244	4488	8976	17952	35904

that  $N \leq k$ . Reporting the number of trainable parameters in a similar table is not readable. It can be computed as  $(c_{in} \cdot c_{out} \cdot k^2)$  for standard kernels,  $(c_{in} \cdot k^2 \cdot N^2)$  for a DCT layer, and  $(c_{in} \cdot k^2 \cdot N(N+1)/2)$  for a  $\lambda$ -truncated DCT layer. The truncation of high frequencies  $N$  or  $\lambda$ , therefore, plays a very important role in reducing the complexity of the networks.

In [Table 4](#), we report similar numbers for a fixed  $c_{in} = 2$ .

**Table 4**

Comparison of number of convolutions between a standard convolutional layer and a DCT layer with a varying number of maximum frequency  $N$ . The number of input channels is set to 2 and the output channels vary from 4 to 64.

$c_{out}$							
	4	8	16	32	64	128	256
Standard conv.							
	8	16	32	64	128	256	512
N	DCT conv.						
5	50	50	50	50	50	50	50
9	162	162	162	162	162	162	162
13	338	338	338	338	338	338	338
17	578	578	578	578	578	578	578
21	882	882	882	882	882	882	882
25	1250	1250	1250	1250	1250	1250	1250
29	1682	1682	1682	1682	1682	1682	1682
33	2178	2178	2178	2178	2178	2178	2178
$\lambda$	DCT conv. $\lambda$ -truncated						
5	30	30	30	30	30	30	30
9	90	90	90	90	90	90	90
13	182	182	182	182	182	182	182
17	306	306	306	306	306	306	306
21	462	462	462	462	462	462	462
25	650	650	650	650	650	650	650
29	870	870	870	870	870	870	870
33	1122	1122	1122	1122	1122	1122	1122

## References

- Depeursinge Adrien, Fageot Julien. Biomedical texture operators and aggregation functions: A methodological review and user's guide. 2017, p. 55–94.
- Varma M, Zisserman A. A statistical approach to texture classification from single images. *Int J Comput Vis* 2005;62(1–2):61–81.
- Mallat Stéphane G. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans Pattern Anal Mach Intell* 1989;11(7):674–93.
- Ojala T, Pietikäinen M, Mäenpää T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 2002;24(7):971–87.
- Simonyan Karen, Zisserman Andrew. Very deep convolutional networks for large-scale image recognition. In: *International conference on learning representations*. 2015, p. 59–68.
- Szegedy Christian, Vanhoucke Vincent, Ioffe Sergey, Shlens Jon, Wojna Zbigniew. Rethinking the inception architecture for computer vision. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, p. 2818–26.
- Ding Xiaohan, Zhang Xiangyu, Han Jungong, Ding Guiguang. 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*. 2022, p. 11963–75.
- Peng Chao, Zhang Xiangyu, Yu Gang, Luo Guiming, Sun Jian. Large kernel matters—improve semantic segmentation by global convolutional network. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, p. 4353–61.
- Luo Wenjie, Li Yujia, Urtasun Raquel, Zemel Richard. Understanding the effective receptive field in deep convolutional neural networks. *Adv Neural Inf Process Syst* 2016;29.
- Chen Yukang, Liu Jianhui, Zhang Xiangyu, Qi Xiaojuan, Jia Jiaya. LargeKernel3D: Scaling up kernels in 3D sparse CNNs. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, p. 13488–98.
- Xie Chengxing, Zhang Xiaoming, Li Linze, Meng Haiteng, Zhang Tianlin, Li Tianrui, Zhao Xiaole. Large kernel distillation network for efficient single image super-resolution. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2023, p. 1283–92.
- Han Shizhong, Meng Zibo, Li Zhiyuan, O'Reilly James, Cai Jie, Wang Xiaofeng, Tong Yan. Optimizing filter size in convolutional neural networks for facial action unit recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, p. 5070–8.
- Liu Ning, Long Yongchao, Zou Changqing, Niu Qun, Pan Li, Wu Hefeng. Adcrowdnet: An attention-injective deformable convolutional network for crowd understanding. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, p. 3225–34.
- Gao Hang, Zhu Xizhou, Lin Steve, Dai Jifeng. Deformable kernels: Adapting effective receptive fields for object deformation. 2019, arXiv preprint [arXiv:1910.02940](https://arxiv.org/abs/1910.02940).
- Bruna Joan, Mallat Stéphane. Invariant scattering convolution networks. *IEEE Trans Pattern Anal Mach Intell* 2013;35(8):1872–86.
- Wang Yunhe, Xu Chang, You Shan, Tao Dacheng, Xu Chao. CNNpack: Packing convolutional neural networks in the frequency domain. In: *Advances in neural information processing systems*. 2016, p. 253–61.
- Qiu Qiang, Cheng Xiuyuan, Sapiro Guillermo, et al. Dcnnet: Deep neural network with decomposed convolutional filters. In: *International conference on machine learning*. PMLR; 2018, p. 4198–207.
- Ulicny Matej, Krylov Vladimir A, Dahyot Rozenn. Harmonic convolutional networks based on discrete cosine transform. *Pattern Recognit* 2022;129:108707.
- Worrall Daniel E, Garbin Stephan J, Turmukhambetov Daniyar, Brostow Gabriel J. Harmonic networks: Deep translation and rotation equivariance. In: *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. 2016, p. 7168–77.
- Oreiller Valentin, Fageot Julien, Andrearczyk Vincent, Prior John O, Depeursinge Adrien. Robust multi-organ nucleus segmentation using a locally rotation invariant bispectral U-Net. In: *Medical imaging with deep learning*. 2021.
- Weiler Maurice, Geiger Mario, Welling Max, Boomsma Wouter, Cohen Taco. 3D steerable CNNs: Learning rotationally equivariant features in volumetric data. In: *NeurIPS*. 2018.
- Andrearczyk Vincent, Fageot Julien, Oreiller Valentin, Montet Xavier, Depeursinge Adrien. Local rotation invariance in 3D CNNs. *Med Image Anal* 2020;65:101756.
- Rao Yongming, Zhao Wenliang, Zhu Zheng, Lu Jiwen, Zhou Jie. Global filter networks for image classification. *Adv Neural Inf Process Syst* 2021;34:980–93.
- Matsoukas Christos, Haslum Johan Fredin, Söderberg Magnus, Smith Kevin. Is it time to replace cnns with transformers for medical images? 2021, arXiv preprint [arXiv:2108.09038](https://arxiv.org/abs/2108.09038).
- Lafarge Maxime W, Bekkers Erik J, Pluim Josien PW, Duits Remco, Veta Mitko. Roto-translation equivariant convolutional networks: Application to histopathology image analysis. *Med Image Anal* 2021;68:101849.
- Ronneberger Olaf, Fischer Philipp, Brox Thomas. U-net: Convolutional networks for biomedical image segmentation. In: *International conference on medical image computing and computer-assisted intervention*. Springer; 2015, p. 234–41.
- Falcão Alexandre X, Stolfi Jorge, de Alencar Lotufo Roberto. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans Pattern Anal Mach Intell* 2004;26(1):19–29.
- Kumar Neeraj, Verma Ruchika, Anand Deepak, Zhou Yanning, Onder Omer Fahri, Tsougenis Efstratios, Chen Hao, Heng Pheng-Ann, Li Jiahui, Hu Zhiqiang, et al. A multi-organ nucleus segmentation challenge. *IEEE Trans Med Imaging* 2019;39(5):1380–91.
- Koboldt Daniel C, Fulton Robert, McLellan Michael, Schmidt Heather, Kalicki-Veizer Joelle, McMichael Joshua, Fulton Lucinda, Dooling David, Ding Li, Mardis Elaine, et al. Comprehensive molecular portraits of human breast tumours. *Nature* 2012;490(7418):61–70.
- Lafarge Maxime W, Pluim Josien PW, Eppenhof Koen AJ, Veta Mitko. Learning domain-invariant representations of histological images. *Front Med* 2019;6:162.
- Mackenro Marc, Niethammer Marc, Marron James S, Borland David, Woosley John T, Guan Xiaojun, Schmitt Charles, Thomas Nancy E. A method for normalizing histology slides for quantitative analysis. In: *2009 IEEE international symposium on biomedical imaging: from nano to macro*. IEEE; 2009, p. 1107–10.