



**HAL**  
open science

# DCT2net: a DCT-based interpretable shallow CNN method for efficient and fast image denoising

Sébastien Herbreteau, Charles Kervrann

► **To cite this version:**

Sébastien Herbreteau, Charles Kervrann. DCT2net: a DCT-based interpretable shallow CNN method for efficient and fast image denoising. IS 2022 - SIAM Conference on Imaging Science, Mar 2022, Berlin, Germany. hal-03926530

**HAL Id: hal-03926530**

**<https://hal.science/hal-03926530>**

Submitted on 6 Jan 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

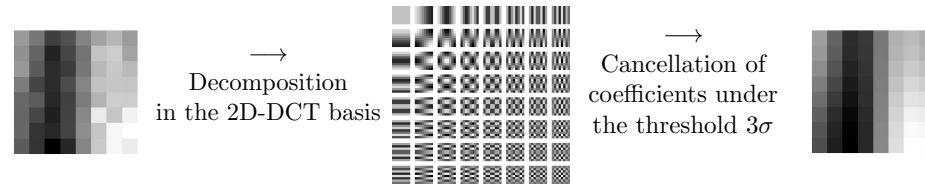
**OVERVIEW**

We propose an **interpretable shallow convolutional neural network** that **generalizes the popular DCT image denoising algorithm** [1]:

- ▷ We show that a DCT denoiser [1] can be seen as a shallow CNN whose weights correspond to the underlying linear transform. Fine-tuning the weights improves considerably the denoising performance.
- ▷ The resulting CNN (DCT2net) is able to handle a wide range of noise levels effectively with a single network thanks to a  $\sigma$ -dependent activation function.
- ▷ Displaying the learned transform gives a direct visual intuition of what the network has learned.
- ▷ To deal with remaining artifacts induced by DCT2net, an original hybrid solution between DCT and DCT2net is proposed.

**DISCRETE COSINE TRANSFORM (DCT) DENOISING**

A DCT denoiser [1] projects each noisy overlapping patch  $y \in \mathbb{R}^n$  to the **2D-DCT basis**, and reconstructs the denoised signal with the transform coefficients higher than the threshold  $3\sigma$ , in absolute value:



Formally, it amounts to estimating every overlapping patch  $x \in \mathbb{R}^n$  by:

$$\hat{x} = P\varphi_\lambda(P^{-1}y) \quad \text{with} \quad \lambda = 3\sigma$$

where  $\varphi_\lambda(t) = t \cdot \mathbb{1}_{\mathbb{R} \setminus [-\lambda, \lambda]}(t)$  is a hard shrinkage function and the columns of  $P \in \mathbb{R}^{n \times n}$  are the basis vectors for the 2D-DCT. The  $n$  estimates of a same pixel are finally aggregated through a weighted average, with respective weight:

$$w = (1 + \|\varphi_\lambda(P^{-1}y)\|_0)^{-1}.$$

**DCT2NET PRINCIPLE**

All operations in DCT algorithm [1] can be interpreted in terms of convolutions with a hard shrinkage function as  $\sigma$ -**dependent activation function** (See Fig. 1). The weights of two first convolutions are to be found in matrices  $P$  and  $P^{-1}$  while the last ones are composed of 0 and 1 for the aggregation step.

To improve the underlying transform, we train our DCT2net on an external dataset composed of  $N$  pairs of noise-free and noisy images  $(x_i, y_i)_{i \in \{1, \dots, N\}}$  corrupted with white Gaussian noise of variance  $\sigma_i^2$ . Training solves:

$$P^* = \arg \min_P \sum_{i=1}^N \|F_P(y_i, \sigma_i) - x_i\|_2^2$$

where  $F_P$  denotes DCT2net.

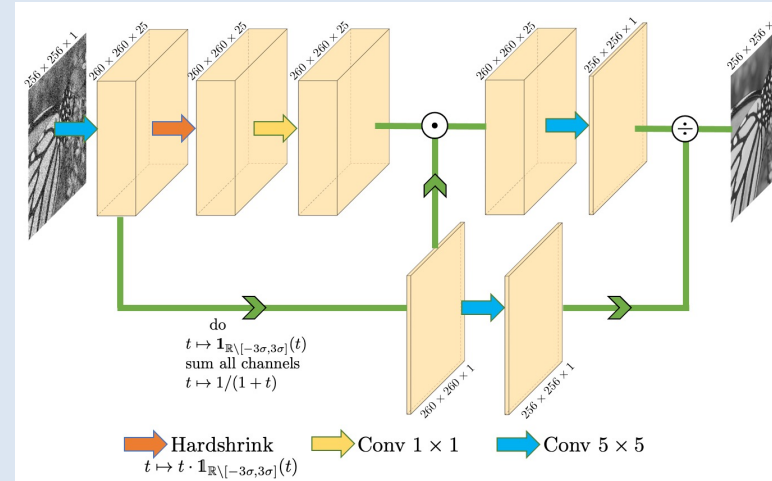


Fig. 1: Architecture of DCT2net for a patch size  $n = 5 \times 5$ . Note that the number of learnable parameters is equal to  $n^2$ : the weights of the first two convolutions being linked by inverse transformation and the last convolutions being frozen with weights in  $\{0, 1\}$  for the aggregation step.

To ease gradient descent during training only,  $t \mapsto \mathbb{1}_{\mathbb{R} \setminus [-\lambda, \lambda]}(t)$  is replaced with the differentiable function:

$$t \mapsto \frac{t^{64}}{t^{64} + \lambda^{64}}$$

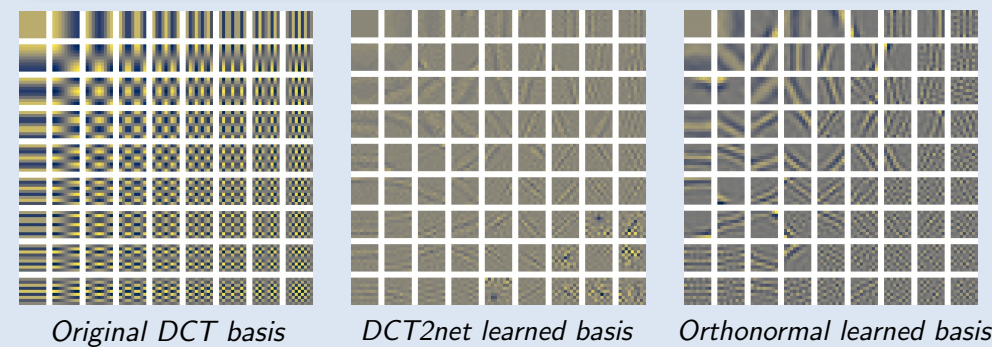
**DISPLAY OF THE LEARNED TRANSFORM**


Fig. 2: Different bases of projection for patches of size  $n = 9 \times 9$ .

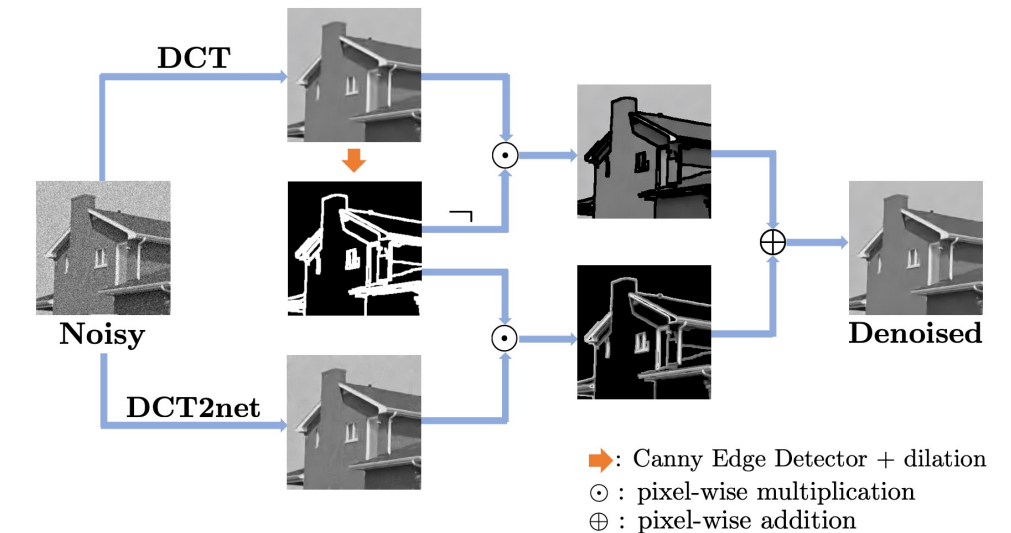
**A FAST AND SHALLOW NETWORK**

Image size	BM3D[2]	PEWA[3]	DnCNN[4]	LKSVD <sub>1,8,256</sub> [5]	DCT2net
256×256	1.73	38.85	0.87 / 0.010	1.15 / 0.020	<b>0.39 / 0.005</b>
512×512	6.65	190.82	3.47 / 0.037	5.78 / 0.082	<b>1.56 / 0.027</b>
1,024×1,024	26.90	803.76	18.35 / 0.145	25.78 / 0.332	<b>5.88 / 0.112</b>

Table 1: Running time in seconds on CPU (left) and GPU (right) when possible.

Methods	DnCNN[4]	LKSVD <sub>1,8,256</sub> [5]	DCT2net
Number of layers	17	5	<b>2</b>
Number of parameters	556,096	35,138	<b>28,561</b>

Table 2: Model complexities comparison of DCT2net with two popular networks.

**DCT2NET MIXED WITH DCT TO REDUCE ARTIFACTS**


- ➔ Canny Edge Detector + dilation
- ⊙: pixel-wise multiplication
- ⊕: pixel-wise addition

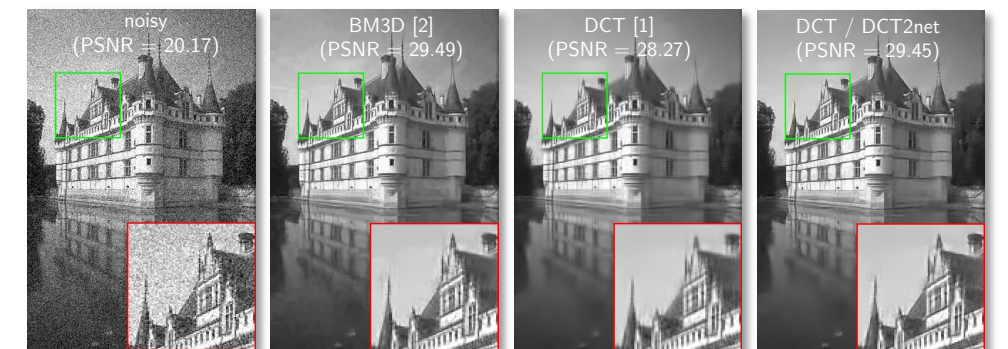
**RESULTS**


Fig. 3: Denoising results on Castle corrupted with white Gaussian noise and  $\sigma = 25$ .

Methods	BM3D[2]	PEWA[3]	DnCNN[4]	LKSVD <sub>1,8,256</sub> [5]	DCT[1]	DCT2net	DCT/DCT2net
$\sigma = 15$	31.07	31.04	<b>31.72</b>	31.33	30.32	31.09	30.97
$\sigma = 25$	28.57	28.52	<b>29.23</b>	28.76	27.76	28.64	28.53
$\sigma = 50$	25.62	25.53	<b>26.23</b>	25.68	24.86	25.68	25.59

Table 3: The average PSNR (dB) results on BSD68 corrupted with Gaussian noise.

**References**

- [1] G. Yu and G. Sapiro, "DCT image denoising: a simple and effective image denoising algorithm," in *Image Processing On Line*, vol. 1, pp. 292–296, 2011.
- [2] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3D transform-domain collaborative filtering," in *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.
- [3] C. Kervrann, "PEWA : Patch-based exponentially weighted aggregation for image denoising," in *Neural Information Processing Systems*, Montreal, Canada, 2014, pp. 2150–2158.
- [4] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: residual learning of deep CNN for image denoising," in *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.
- [5] M. Scetbon, M. Elad, and P. Milanfar, "Deep K-SVD Denoising," in *IEEE Transactions on Image Processing*, vol. 30, pp. 5944–5955, 2021.
- [6] J. Canny, "A computational approach to edge detection," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, 1986.