function approximation [41], signal representation and classification [34]. In these early works, the authors apply shallow networks to search the optimal wavelet in wavelet parameter domain. Recently, this method is utilized with deeper network for image classification, but the network is difficult to train because of the significant amount of computational cost [7]. ScatNet [5] cascades wavelet transform with nonlinear modulus and average pooling, to extract a translation invariant feature robust to deformations and preserve high-frequency information for image classification. The authors introduce ScatNet when they explore from mathematical and algorithmic perspective how to design the optimal deep network. Compared with the CNNs of the same period, ScatNet gets better performance on the handwritten digit recognition and texture discrimination tasks. However, due to the strict mathematical assumptions, ScatNet can not be easily transferred to other tasks.

In deep learning, wavelets commonly play the roles of image preprocessing or postprocessing [17, 23, 32, 39]. Meanwhile, researchers try to introduce wavelet transforms into the design of deep networks in various tasks [22, 35, 11, 37], by taking wavelet transforms as sampling operations. Multi-level Wavelet CNN (MWCNN) proposed in [22] integrates Wavelet Package Transform (WPT) into the deep network for image restoration. MWCNN concatenates the low-frequency and high-frequency components of the input feature map, and processes them in a unified way, while the data distribution in these components significantly differs from each other. Convolutional-Wavelet Neural Network (CWNN) proposed in [11] applies dual-tree complex wavelet transform (DT-CWT) to suppress the noise and keep the structures for extracting robust features from SAR images. The architecture of CWNN contains only two convolution layers. While DT-CWT is redundant, CWNN takes as its down-sampling output the average value of the multiple components output from DT-CWT. Wavelet pooling proposed in [35] is designed using a two-level DWT. Its back-propagation performs a one-level DWT and a two-level IDWT, which does not follow the mathematical principle of gradient. The authors test their method on various dataset (MNIST [20], CIFAR-10 [18], SHVN [27], and KDEF [24]). However, their network architectures contain only four or five convolutional layers. The authors do not study systematically the potential of the method on standard image dataset like ImageNet [8]. Recently, the application of wavelet transform in image style transfer is studied in [37]. In above works, the authors evaluate their methods with only one or two wavelets, due to the absence of the general wavelet transform layers.

3. Our method

Our method is trying to apply wavelet transforms to improve the down-sampling operations in deep networks. We firstly design the general DWT and IDWT layers.

3.1. DWT and IDWT layers

The key issues in designs of DWT and IDWT layers are the data forward and backward propagations. Although the following analysis is for orthogonal wavelet and 1D signal, it can be generalized to other wavelets and 2D/3D signal with only slight changes.

Forward propagation For a 1D signal $\mathbf{s} = \{s_j\}_{j\in\mathbb{Z}}$, DWT decomposes it into its low-frequency component $\mathbf{s}_1 = \{s_{1k}\}_{k\in\mathbb{Z}}$ and high-frequency component $\mathbf{d}_1 = \{d_{1k}\}_{k\in\mathbb{Z}}$, where

$$\begin{cases} s_{1k} = \sum_j l_{j-2k} s_j, \\ d_{1k} = \sum_j h_{j-2k} s_j, \end{cases} \tag{1}$$

and $\mathbf{l} = \{l_k\}_{k\in\mathbb{Z}}, \mathbf{h} = \{h_k\}_{k\in\mathbb{Z}}$ are the low-pass and high-pass filters of an orthogonal wavelet. According to Eq. (1), DWT consists of filtering and down-sampling.

Using IDWT, one can reconstruct $\mathbf{s}$ from $\mathbf{s}_1, \mathbf{d}_1$, where

$$s_j = \sum_k \left(l_{j-2k} s_{1k} + h_{j-2k} d_{1k}\right). \tag{2}$$

In expressions with matrices and vectors, Eq. (1) and Eq. (2) can be rewritten as

$$\mathbf{s}_1 = \mathbf{L}\mathbf{s}, \quad \mathbf{d}_1 = \mathbf{H}\mathbf{s}, \tag{3}$$

$$\mathbf{s} = \mathbf{L}^T\mathbf{s}_1 + \mathbf{H}^T\mathbf{d}_1, \tag{4}$$

where

$$\mathbf{L} = \begin{pmatrix} \cdots & \cdots & \cdots & & & & \\ \cdots & l_{-1} & l_0 & l_1 & \cdots & & \\ & & \cdots & l_{-1} & l_0 & l_1 & \cdots \\ & & & & & \cdots & \cdots \end{pmatrix}, \tag{5}$$

$$\mathbf{H} = \begin{pmatrix} \cdots & \cdots & \cdots & & & & \\ \cdots & h_{-1} & h_0 & h_1 & \cdots & & \\ & & \cdots & h_{-1} & h_0 & h_1 & \cdots \\ & & & & & \cdots & \cdots \end{pmatrix}. \tag{6}$$

For 2D signal $\mathbf{X}$, the DWT usually do 1D DWT on its every row and column, i.e.,

$$\mathbf{X}_{ll} = \mathbf{L}\mathbf{X}\mathbf{L}^T, \tag{7}$$

$$\mathbf{X}_{lh} = \mathbf{H}\mathbf{X}\mathbf{L}^T, \tag{8}$$

$$\mathbf{X}_{hl} = \mathbf{L}\mathbf{X}\mathbf{H}^T, \tag{9}$$

$$\mathbf{X}_{hh} = \mathbf{H}\mathbf{X}\mathbf{H}^T, \tag{10}$$

and the corresponding IDWT is implemented with

$$\mathbf{X} = \mathbf{L}^T\mathbf{X}_{ll}\mathbf{L} + \mathbf{H}^T\mathbf{X}_{lh}\mathbf{L} + \mathbf{L}^T\mathbf{X}_{hl}\mathbf{H} + \mathbf{H}^T\mathbf{X}_{hh}\mathbf{H}. \tag{11}$$

Backward propagation For the backward propagation of DWT, we start from Eq. (3) and differentiate it,

$$\frac{\partial \mathbf{s}_1}{\partial \mathbf{s}} = \mathbf{L}^T, \quad \frac{\partial \mathbf{d}_1}{\partial \mathbf{s}} = \mathbf{H}^T. \tag{12}$$