Joint Learning of Fully Connected Network Models in Lifting Based Image Coders

Tassnim Dardouri, *Student Member, IEEE*, Mounir Kaaniche, *Senior Member, IEEE*, Amel Benazza-Benyahia, Gabriel Dauphin, and Jean-Christophe Pesquet, *Fellow, IEEE*

***Abstract*— The optimization of prediction and update operators plays a prominent role in lifting-based image coding schemes. In this paper, we focus on learning the prediction and update models involved in a recent Fully Connected Neural Network (FCNN)-based lifting structure. While a straightforward approach consists in separately learning the different FCNN models by optimizing appropriate loss functions, jointly learning those models is a more challenging problem. To address this problem, we first consider a statistical model-based entropy loss function that yields a good approximation to the coding rate. Then, we develop a multi-scale optimization technique to learn all the FCNN models simultaneously. For this purpose, two loss functions defined across the different resolution levels of the proposed representation are investigated. While the first function combines standard prediction and update loss functions, the second one aims to obtain a good approximation to the rate-distortion criterion. Experimental results carried out on two standard image datasets, show the benefits of the proposed approaches in the context of lossy and lossless compression.**

***Index Terms*— Lifting schemes, adaptive wavelets, image coding, neural networks, joint learning, optimization.**

I. Introduction

LIFTING Scheme (LS), also known as the second generation of wavelets, was found to be a powerful tool in image and video processing [1], [2]. LS has been retained in the JPEG2000 image compression standard [3] due to its many advantages with respect to classical methods for constructing wavelets based on a discrete filter bank implementation. These advantages include implementation simplicity and perfect reconstruction property. Since the inclusion of LS in JPEG2000 image compression standard, several research publications have shown the benefit of LS for the coding of other types of data like video, stereo images, holograms, etc [4], [5], [6]. Recently, invertible neural networks inspired by the lifting scheme have been developed for various image processing tasks such as classification [7], denoising [8], and coding [9], [10]. The latter will be the focus of this paper.

Lifting schemes are basically composed of prediction and update steps that aim to generate detail (i.e., high frequency) and approximation (i.e., low frequency) wavelet subbands, respectively [11], [12]. For instance, for image decomposition, two-dimensional non separable lifting structures can be used to better capture the 2D characteristics of the input image. A typical non separable LS consists of three prediction steps followed by an update step. This yields three detail subbands oriented diagonally, vertically, and horizontally as well as one approximation subband [13], [14]. Generally, the coding performance of these LS depend on the design of the prediction and update operators. As a result, many efforts have been devoted to optimizing such operators while making the wavelet decomposition adaptable to the image contents [14], [15], [16], [17]. In fact, the optimization of the prediction operator is often achieved by minimizing the ℓ_2-norm of the detail coefficients [18]. To further promote sparsity of the wavelet coefficients, ℓ_1 and weighted ℓ_1-based minimization techniques have also been investigated in [19]. In addition, an entropy measure has also been used in [16] and [20] for the same purpose. The corresponding optimization problem is solved empirically using the Nelder-Mead simplex algorithm. While the optimization of the predictor has been widely studied, optimizing the update filter is less obvious and few studies have pursued this direction [18], [21], [22]. These studies rely on two main techniques. The first one consists in minimizing the reconstruction error while synthesizing the reconstructed image from the approximation coefficients [18], [21]. However, this technique results in a complex linear system of equations. To overcome this issue, the second technique aims to minimize the error between the approximation subband and the decimated version of the output of an ideal loss-pass filter applied to the input image [22].

Motivated by the success of neural networks and their advantages in achieving accurate nonlinear approximation, the prediction and update lifting stages have been recently implemented using Convolutional Neural Network (CNN) [23], [24] and Fully Connected Neural Network (FCNN) [10], [25]. These neural networks-based lifting architectures will be further described and discussed in Section II. In addition to this

Manuscript received 13 February 2023; revised 21 July 2023 and 3 October 2023; accepted 31 October 2023. Date of publication 21 November 2023; date of current version 8 December 2023. This work was supported in part by the Agence nationale de la recherche (ANR) Research and Teaching Chair BRIDinG thE gAp Between iterative proximaL methods and nEural networks (BRIDGEABLE) in Artificial Intelligence. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Zhenzhong Chen. *(Corresponding author: Tassnim Dardouri.)*

Tassnim Dardouri is with Novelis, 75012 Paris, France (e-mail: tdardouri@novelis.io).

Mounir Kaaniche and Gabriel Dauphin are with L2TI, UR 3043, Université Sorbonne Paris Nord, 93430 Villetaneuse, France (e-mail: mounir.kaaniche@univ-paris13.fr; gabriel.dauphin@univ-paris13.fr).

Amel Benazza-Benyahia is with SUP'COM, LR11TIC01, COSIM Laboratory, University of Carthage, El Ghazala, Carthage 2083, Tunisia (e-mail: benazza.amel@supcom.rnu.tn).

Jean-Christophe Pesquet is with Centre de Vision Numérique, CentraleSupélec, Inria, Université Paris-Saclay, 91190 Gif-sur-Yvette, France (e-mail: jean-christophe@pesquet.eu).

Digital Object Identifier 10.1109/TIP.2023.3333279

class of methods, other studies have sought to improve DCT (Discrete Cosine Transform) and Discrete Wavelet Transform (DWT)-based coding schemes [26], [27], [28]. In fact, in [26], a CNN architecture is exploited to develop a DCT-like transform for image coding. In [27], the authors apply a DWT to the original image, and the generated wavelet subbands are fed into a CNN to produce the final detail coefficients. Other research efforts have been devoted to intra-prediction coding techniques using both CNN and FCNN [29], [30], [31]. For instance, in [30], the authors proposed to apply the FCNN to small image blocks and the CNN to large blocks. Moreover, most of the remaining existing NN-based image compression methods follow the same kind of procedure. First, a nonlinear analysis transform is applied to the input image. The generated feature maps are then quantized and entropy encoded. Finally, a nonlinear synthesis transform is performed to reconstruct the image. These methods are optimized using an end-to-end procedure and mainly differ in the employed NN models [32], [33], [34], [35] and/or the loss function used for training [36], [37], [38]. In this category of learned image compression techniques, context model for entropy coding is also investigated to further boost the rate-distortion performance [37], [39], [40]. It is worth pointing out that most of the developed NN-based image compression methods are devoted to lossy compression [41] and only a few studies are dealing with lossless compression [42], [43], [44].

The objective of this paper is to develop new learning techniques for optimizing neural network models in lifting scheme-based image coders. More precisely, we focus on a recent non separable lifting architecture involving three FCNN-based prediction steps followed by an FCNN-based update one [10]. While these different FCNN models have been *separately* learned in [10], we propose here to investigate *joint* learning approaches to find the optimal FCNN models. Note that a preliminary version of this work has been presented in [25]. For instance, unlike [10] and [25] where ℓ_2- and ℓ_1-norm based loss functions are employed, we resort to an entropy based loss function in this paper. The loss function is grounded on a Generalized Gaussian probabilistic model which has been widely employed for wavelet coefficients modeling. Moreover, this work aims to learn both the FCNN prediction and update models *simultaneously*, while the joint learning of three FCNN prediction models is achieved independently of that of the FCNN update one in [25]. To this end, we propose a multi-level optimization technique. This technique consists in interpreting the lifting-based multiresolution decomposition as a full architecture whose involved FCNN models are globally learned at the same time through a *unique* loss function. In this respect, two new loss functions will be investigated. While the first one resorts to a weighted sum of the loss functions used to optimize the prediction and update stages, the second one aims to obtain a good approximation of rate-distortion functions.

The remainder of this paper is organized as follows. In Section II, we describe the related neural networks-based lifting schemes and then focus on the FCNN-based structure we recently proposed. Our statistical model-based entropy loss function is introduced in Section III. The developed learning approaches for the different FCNN models are described in Section IV. Finally, extensive experiments are shown and discussed in Section V, and our conclusions are presented in Section VI.

II. Fully Connected Neural Network Based Lifting Architecture

A. Related Neural Networks-Based Lifting Schemes

The use of neural networks for the design of lifting based image coding schemes have been mainly addressed in recent works conducted by Ma et al. [23], [24] and Dardouri et al. [10], [25]. In [23], the authors have considered a separable lifting structure where the prediction stage is achieved using a CNN model and the update one is performed by a mean operation. The corresponding network parameters are then learned by optimizing a distortion criterion. The latter scheme (called *iwave*), has been extended in [24] (named *iwave*++) by applying CNNs to both prediction and update stages and optimizing the architecture in an end-to-end fashion using a rate-distortion-based loss function. It must be emphasized that the use of neural networks has been investigated in three different modules [24]: lifting decomposition, entropy coding, and post-processing, resulting in a high computational complexity. Four variants have been developed. While the first one is dedicated to lossless compression, the second and third ones are designed for lossy coding and consider single and multiple NN models, respectively. The last variant is a universal scheme which uses a single model for both lossy and lossless compression. However, the latter suffers from two main drawbacks. First, while the best lossy compression performance is obtained with the multi-model configuration requiring a separate NN model for each point of the R-D curve, the universal scheme results in a significant performance drop. Moreover, *iwave* and *iwave*++ rely on the concept of a one-dimensional (1D) LS-based decomposition, which yield an increase of the number of employed NN models, and so the number of parameters, in the whole multiresolution architecture.

To alleviate these shortcomings, we have proposed in [10] and [25] to design a single FCNN model for both lossy and lossless compression while focusing on a Non-Separable Lifting Scheme (NSLS) [19] which better captures the 2D characteristics of the data and reduces the number of lifting stages performed in the image decomposition. More precisely, both prediction and update steps are achieved using an FCNN. To this end, new loss functions, defined in the wavelet transform domain, for learning the FCNN prediction and update models have been proposed. While all the FCNN models are learned in a separate manner in [10], joint learning restricted to prediction models is developed in [25].

B. FCNN Based Wavelet Representation

To perform wavelet decomposition, we have retained the conventional NSLS composed of three prediction steps followed by an update one. In conventional image coders, these steps are often achieved using linear operators. To go

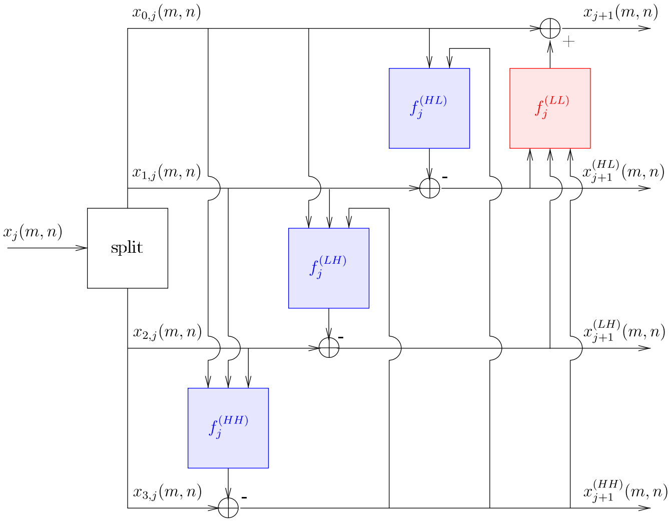


Fig. 1. Analysis structure of the FCNN-based NSLS architecture.

beyond these linear models and obtain more accurate non-linear approximation properties, we have recently proposed performing the prediction and update using neural networks, and more specifically FCNN models. The analysis structure of this new FCNN based NSLS is shown in Fig. 1. More precisely, x_j denotes the approximation subband at resolution level j where $x_0 = x$ represents the input image. In the first split stage, the input x_j is decomposed into four subsets of samples denoted by

$$\begin{cases} x_{0,j}(m,n) = x_j(2m,2n), \\ x_{1,j}(m,n) = x_j(2m,2n+1), \\ x_{2,j}(m,n) = x_j(2m+1,2n), \\ x_{3,j}(m,n) = x_j(2m+1,2n+1). \end{cases} \tag{1}$$

Then, three prediction stages, based on three FCNN models $f_j^{(HH)}$, $f_j^{(LH)}$ and $f_j^{(HL)}$, are applied to generate the diagonal detail subband $x_{j+1}^{(HH)}$, the vertical subband $x_{j+1}^{(LH)}$, and the horizontal subband $x_{j+1}^{(HL)}$, respectively. These detail coefficients are computed as follows:

$$\begin{aligned} &\forall\ o \in \{HH, LH, HL\}, \\ x_{j+1}^{(o)}(m,n) &= x_{i,j}(m,n) - \hat{x}_{i,j}(m,n) \\ &= x_{i,j}(m,n) - f_j^{(o)}(\tilde{\mathbf{x}}_j^{(o)}(m,n)) \end{aligned} \tag{2}$$

where, for each $i \in \{1,2,3\}$, $x_{i,j}(m,n)$ is the sample to be predicted, $\tilde{\mathbf{x}}_j^{(o)}(m,n)$ is the input reference vector containing the samples used to generate the detail coefficients $x_{j+1}^{(o)}$, and $\hat{x}_{i,j}(m,n)$ is the predicted value that corresponds to the output of the FCNN model $f_j^{(o)}(\tilde{\mathbf{x}}_j^{(o)}(m,n))$.

As shown in Fig. 2, the reference vector $\tilde{\mathbf{x}}_j^{(o)}(m,n)$ is first fed to the input layer of an FCNN model. Then, a few hidden layers with various dimensions (i.e., number of neurons) are employed. The output values of these neurons are computed based on a linear combination (with bias) followed by a nonlinear activation function. Finally, the output layer, with a single neuron, generates the predicted value $\hat{x}_{i,j}(m,n)$ based on a linear combination of the neuron values associated with the last hidden layer.

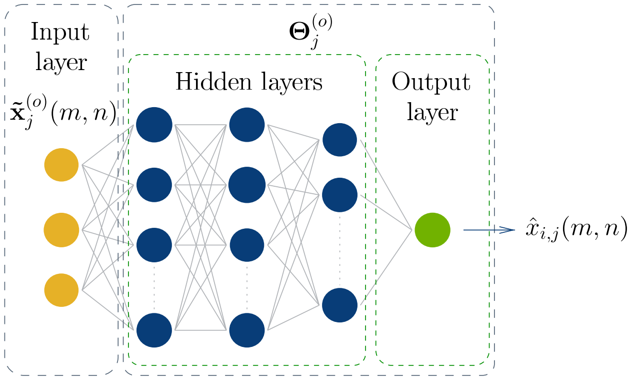


Fig. 2. FCNN-based prediction stage.

Following the three prediction steps, an update step using an FCNN model $f_j^{(LL)}$ is performed to produce the approximation coefficients x_{j+1}:

$$\begin{aligned} x_{j+1}(m,n) &= x_{0,j}(m,n) + \hat{t}_j(m,n) \\ &= x_{0,j}(m,n) + f_j^{(LL)}(\tilde{\mathbf{x}}_{j+1}(m,n)) \end{aligned} \tag{3}$$

where $\tilde{\mathbf{x}}_{j+1}(m,n)$ is the input reference vector containing the diagonal, vertical, and horizontal detail coefficients.

Similar to the FCNN-based prediction stage, $\tilde{\mathbf{x}}_{j+1}(m,n)$ serves as an input vector for the FCNN update model $f_j^{(LL)}$. A stack of hidden layers followed by an output layer with a single neuron are then used to produce $\hat{t}_j(m,n)$ and deduce the approximation coefficients x_{j+1} according to (3).

C. Independent Learning Approaches

To learn the four FCNN models, $f_j^{(o)}$ with $o \in \{HH, LH, HL, LL\}$, a simple approach, adopted in [10], trains each model defined at a given resolution level j. We recall the two optimization techniques which have been employed for learning the prediction and update models.

1) Learning FCNN-Based Prediction Models: Let $\Theta_j^{(o)}$, with $o \in \{HH, LH, HL\}$ denote the vector of parameters associated with the FCNN model $f_j^{(o)}$. Thus, for each detail subband $x_{j+1}^{(o)}$, the vector $\Theta_j^{(o)}$ is learned by performing several forward and backward propagation passes while minimizing a loss function. In this manner, the Mean Square Error (MSE) loss function has been widely used in regression tasks. As a result, we proposed in [10] to train each FCNN-based prediction model using the ℓ_2-norm of the prediction error between the target pixels $x_{i,j}(m,n)$ and the predicted ones $\hat{x}_{i,j}(m,n)$:

$$\begin{aligned} &\forall\ o \in \{HH, LH, HL\}, \\ &\widetilde{\mathcal{L}}^{(\mathrm{p})}(\Theta_j^{(o)}) = \frac{1}{M_j N_j} \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \big(x_{i,j}(m,n) - \hat{x}_{i,j}(m,n)\big)^2 \end{aligned} \tag{4}$$

where M_j and N_j represent the dimensions of the image x_j divided by 2. Note that the superscript 'p' is used to emphasize that the above loss function is specific to the prediction model.

To optimize this loss function, a Mini-Batch Gradient Descent (MBGD) algorithm has been employed [45]. For instance, the gradient of the loss function is computed for each mini-batch and the model parameters are then updated from one mini-batch to the next one. By repeating this process over several epochs until convergence of the algorithm, the optimal weights $\Theta_j^{(o)}$ are obtained. The optimal weights are finally applied to the test images to find the predicted pixels $\hat{x}_{i,j}$ and produce the detail coefficients $x_{j+1}^{(o)}$ using (2).

2) Learning the FCNN-Based Update Model: Once the three FCNN prediction models are learned and their respective detail subbands are produced, one can focus on learning the FCNN update model. However, since the generation of the approximation coefficients is quite different from that of the detail ones, a more appropriate loss function should be employed for the update stage. In this respect, we proposed to use a simple and efficient update optimization design method which consists in minimizing the error between the approximation subband and the decimated version of the output of an ideal low pass filter (whose impulse response will be denoted by $\tilde{h}$) applied to the input x_j [22]. This error is given by

$$\begin{aligned} e_j(m,n) &= y_{j+1}(m,n) - x_{j+1}(m,n) \\ &= y_{j+1}(m,n) - x_{0,j}(m,n) - f_j^{(LL)}(\tilde{\mathbf{x}}_{j+1}(m,n)) \end{aligned} \tag{5}$$

where

$$y_{j+1}(m,n) = (\tilde{h} * x_j)(2m, 2n). \tag{6}$$

Thanks to (5), the update learning task can be seen as a prediction learning task that aims to predict, from the input reference vector $\tilde{\mathbf{x}}_{j+1}(m,n)$, the target signal $t_j(m,n)$ defined as

$$t_j(m,n) = y_{j+1}(m,n) - x_{0,j}(m,n). \tag{7}$$

Therefore, similar to (4), the FCNN-based update model is trained by minimizing the MSE between $t_j(m,n)$ and $\hat{t}_j(m,n) = f_j^{(LL)}(\tilde{\mathbf{x}}_{j+1}(m,n))$. Thus, the loss function used for training the FCNN update model is given by

$$\begin{aligned} &\widetilde{\mathcal{L}}^{(\mathrm{u})}(\Theta_j^{(LL)}) \\ &= \frac{1}{M_j N_j} \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \left(y_{j+1}(m,n) - x_{0,j}(m,n) - \hat{t}_j(m,n)\right)^2 \end{aligned} \tag{8}$$

where the superscript 'u' is employed to refer to the loss function used with the update model.

Once the update network is trained, its optimal vector of parameters $\Theta_j^{(LL)}$ is employed to compute the output $\hat{t}_j(m,n)$ from the input vector $\tilde{\mathbf{x}}_{j+1}(m,n)$ and generate the approximation coefficients using (3).

For the test phase, and in order to generate integer wavelet coefficients, which are mandatory for lossless compression, a rounding operator is applied to the second terms (i.e., $f_j^{(o)}(\cdot)$) of (2) and (3).

D. Analysis of the FCNN-Based Lifting Structure

The employed loss functions and learning strategy present some advantages and drawbacks. One of these advantages is that the defined loss functions are computed in the transform domain and so, they do not require to perform image reconstruction as often considered in the existing learned image compression methods. Moreover, the adopted learning strategy can be seen as the straightforward solution which consists in training the different FCNN models in a separate manner. This allows to learn some models (in particular those related to the second and third prediction stages) in parallel.

However, such an independent learning approach does not take into account two main aspects inherent to NSLS based coding schemes. The first aspect concerns the dependencies existing between these models at a given resolution level j. As it can be seen in Fig. 1, the output of the first FCNN prediction model, and more specifically the diagonal detail signal $x_{j+1}^{(HH)}$, is used as the reference signal (i.e., as an input) of the second and third FCNN prediction models to produce the vertical and horizontal detail signals. The second aspect is related to the multiresolution form of the NSLS architecture. A decomposition carried out over J resolution levels is iteratively performed, and yields one approximation subband and $3J$ detail subbands. This means that the coefficients computed at a given level will impact those generated at coarser levels.

For this reason, we propose in this paper to resort to novel joint learning approaches while investigating different loss functions, as it will be described in Sections III and IV.

III. Proposed Statistical Model Based Loss Function

While the mean square prediction error has been widely used as a loss function for learning predictive models, we propose here to explore an entropy-based loss function. This choice is motivated by the fact that the entropy represents a good approximation to the final bitrate [46].

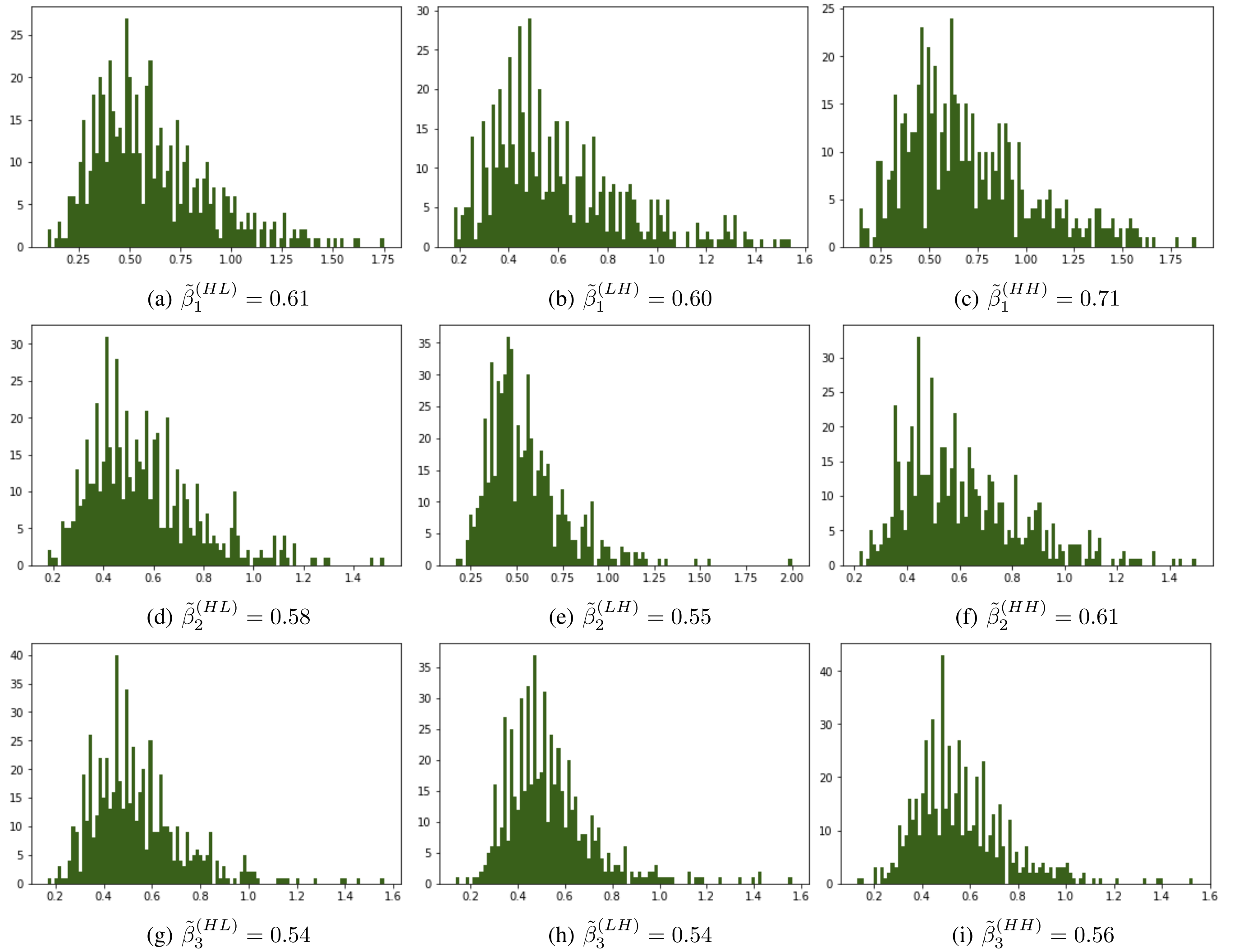
A. Statistical Model

The analysis of wavelet coefficients and their probability distributions have often been exploited in various image processing tasks including denoising and compression. For instance, the Generalized Gaussian Distribution (GGD) has been extensively used to model the subband coefficients $x_{j+1}^{(o)}$ of a given wavelet representation. These coefficients can be seen as realizations of a random variable $X_{j+1}^{(o)}$ whose probability density function $g_{j+1}^{(o)}$ is given by

$$\forall o \in \{HH, LH, HL\}, \quad \forall \xi \in \mathbb{R},$$

$$g_{j+1}^{(o)}(\xi; \alpha_{j+1}^{(o)}, \beta_{j+1}^{(o)}) = \frac{\beta_{j+1}^{(o)}}{2\alpha_{j+1}^{(o)} \Gamma(\frac{1}{\beta_{j+1}^{(o)}})} e^{-\left(\frac{|\xi|}{\alpha_{j+1}^{(o)}}\right)^{\beta_{j+1}^{(o)}}} \tag{9}$$

where Γ is the Gamma function, $\alpha_{j+1}^{(o)} \in]0, +\infty[$ is the scale parameter, and $\beta_{j+1}^{(o)} \in]0, +\infty[$ is the shape parameter of the GGD. The particular case when $\beta_{j+1}^{(o)} = 2$ (resp. $\beta_{j+1}^{(o)} = 1$)



(a) $\tilde{\beta}_1^{(HL)} = 0.61$

(b) $\tilde{\beta}_1^{(LH)} = 0.60$

(c) $\tilde{\beta}_1^{(HH)} = 0.71$

(d) $\tilde{\beta}_2^{(HL)} = 0.58$

(e) $\tilde{\beta}_2^{(LH)} = 0.55$

(f) $\tilde{\beta}_2^{(HH)} = 0.61$

(g) $\tilde{\beta}_3^{(HL)} = 0.54$

(h) $\tilde{\beta}_3^{(LH)} = 0.54$

(i) $\tilde{\beta}_3^{(HH)} = 0.56$

Fig. 3. Distribution of $\beta_{j+1}^{(o)}$ values for the horizontal (first column), vertical (second column) and diagonal (third column) detail subbands at the first three resolution levels (from top to bottom).

corresponds to the Gaussian distribution (resp. the Laplacian one).

Let us also recall that the use of ℓ_2 (resp. ℓ_1)-norm based loss function suggests that the data distribution follows such a Gaussian (resp. Laplacian) model. However, multiresolution representations have the advantage of producing sparse coefficients whose distribution shape parameters $\beta_{j+1}^{(o)}$ are smaller than 1. To confirm this property, we have conducted a distribution analysis of the detail coefficients, and more specifically the shape parameters $\beta_{j+1}^{(o)}$ of their associated GGD models, for a large set of samples extracted from the popular CLIC database described in Section V. Note that the parameters of the GGD models are estimated by applying the maximum likelihood technique [47] to the subband coefficients produced by the independent MSE-based learned FCNN model (see Section II-C). Fig. 3 illustrates the distribution of the estimated $\beta_{j+1}^{(o)}$ values obtained with the three detail subbands at the first three resolution levels. We also provide the average values

$$\tilde{\beta}_{j+1}^{(o)} = \frac{1}{K}\sum_{k=1}^{K} \beta_{j+1}^{(o,k)} \tag{10}$$

where K is the number images. Thus, it can be observed that typical values of the shape parameters $\beta_{j+1}^{(o)}$ range from 0.2 to 1 and their mean values are around 0.6. Moreover, these values become smaller at coarser levels.

Based on this analysis, instead of using a given value of $\beta_{j+1}^{(o)}$ (i.e., $\beta_{j+1}^{(o)} = 2$ or $\beta_{j+1}^{(o)} = 1$) as is often considered in previous studies, we propose here to adaptively select the values of $\beta_{j+1}^{(o)}$ that depend on the subband orientation o as well as the resolution level j.

B. Entropy-Based Loss Function

Let $\bar{x}_{j+1}^{(o)}(m, n)$ be the quantized coefficients using a uniform scalar quantizer with a quantization step $q_{j+1}^{(o)}$. These coefficients can also be seen as realizations of a random variable $\bar{X}_{j+1}^{(o)}$. At high bitrates, it has been shown in [46] that the discrete entropy of $\bar{X}_{j+1}^{(o)}$ can be approximated as follows:

$$H(\bar{X}_{j+1}^{(o)}) \approx h(X_{j+1}^{(o)}) - \log_2(q_{j+1}^{(o)}) \tag{11}$$

where $h(X_{j+1}^{(o)})$ is the differential entropy of the variable $X_{j+1}^{(o)}$.

Thus, it can be seen that the discrete entropy of the quantized source is (up to an additive constant) approximately equal to the differential entropy of the original (i.e., non-quantized) source. For such a source following a GGD, the law of large numbers yields the following expression of the differential entropy:

$$h(X_{j+1}^{(o)}) \approx \frac{1}{M_j N_j \ln(2)(\alpha_{j+1}^{(o)})^{\beta_{j+1}^{(o)}}} \sum_{m=1}^{M_j}\sum_{n=1}^{N_j} \left| x_{j+1}^{(o)}(m,n)\right|^{\beta_{j+1}^{(o)}} + \log_2\Big(\frac{2\alpha_{j+1}^{(o)}\Gamma\big(\frac{1}{\beta_{j+1}^{(o)}}\big)}{\beta_{j+1}^{(o)}}\Big). \tag{12}$$

Therefore, with the ultimate goal of approximating the bitrate of the subbands to be generated, we propose to use the differential entropy of the detail coefficients as a learning loss for the different FCNN-based prediction models.

In this context, a possible choice for setting the $\beta_{j+1}^{(o)}$ and $\alpha_{j+1}^{(o)}$ parameters consists of using values $\tilde{\beta}_{j+1}^{(o)}$ and $\tilde{\alpha}_{j+1}^{(o)}$ estimated from K training images. Their subband coefficients are computed by the independent MSE-based learned FCNN model. By using an averaged maximum likelihood estimator, we obtain

$$\tilde{\alpha}_{j+1}^{(o)} = \frac{1}{K}\sum_{k=1}^{K}\Big(\frac{\tilde{\beta}_{j+1}^{(o)}}{M_j N_j}\sum_{m=1}^{M_j}\sum_{n=1}^{N_j}|\tilde{x}_{j+1}^{(o,k)}(m,n)|^{\tilde{\beta}_{j+1}^{(o)}}\Big)^{1/\tilde{\beta}_{j+1}^{(o)}}, \tag{13}$$

where $\tilde{x}_{j+1}^{(o,k)}$ is the detail subband of the k-th training image whose GGD shape parameter is $\tilde{\beta}_{j+1}^{(o)}$. The expression of $\tilde{\beta}_{j+1}^{(o)}$ is still given by (10). By omitting constant terms, this approach leads to the following loss function, which in turn is used to learn the FCNN prediction weight parameters $\Theta_j^{(o)}$:

$$\forall\, o \in \{HH, LH, HL\}, \quad \mathcal{L}_{j,o}^{(\mathrm{p},1)}(\Theta_j^{(o)}) = \frac{1}{M_j N_j \ln(2)(\tilde{\alpha}_{j+1}^{(o)})^{\tilde{\beta}_{j+1}^{(o)}}}\sum_{m=1}^{M_j}\sum_{n=1}^{N_j}\left|x_{j+1}^{(o)}(m,n)\right|^{\tilde{\beta}_{j+1}^{(o)}}. \tag{14}$$

This loss function provides a natural extension of both ℓ_1 and ℓ_2 losses. It relies however on a relatively rough averaged estimate of $\alpha_{j+1}^{(o)}$.

A second approach for estimating the $\alpha_{j+1}^{(o)}$ parameter relies on the maximum likelihood estimate that would be obtained from the generated subband coefficients:

$$\hat{\alpha}_{j+1}^{(o)} = \Big(\frac{\beta_{j+1}^{(o)}}{M_j N_j}\sum_{m=1}^{M_j}\sum_{n=1}^{N_j}|x_{j+1}^{(o)}(m,n)|^{\beta_{j+1}^{(o)}}\Big)^{1/\beta_{j+1}^{(o)}}. \tag{15}$$

Although this estimate cannot be practically calculated, its expression can be injected into (12), thus leading to the definition of an alternative form of the loss function for minimizing the entropy of the wavelet coefficients:

$$\forall\, o \in \{HH, LH, HL\}, \quad \mathcal{L}_{j,o}^{(\mathrm{p},2)}(\Theta_j^{(o)}) = \frac{1}{\tilde{\beta}_{j+1}^{(o)}}\log_2\Big(\frac{1}{M_j N_j}\sum_{m=1}^{M_j}\sum_{n=1}^{N_j}\left|x_{j+1}^{(o)}(m,n)\right|^{\tilde{\beta}_{j+1}^{(o)}}\Big). \tag{16}$$

Because of the behaviour of the $\log_2$ function around 0, this loss function has however a tendency to be sensitive to the dynamics of the subband coefficients.

To benefit from the advantages of the two previous approaches, we propose to define the loss function as the arithmetic mean of the two previous expressions in (14) and (16), i.e.

$$\forall\, o \in \{HH, LH, HL\}, \quad \mathcal{L}_{j,o}^{(\mathrm{p})}(\Theta_j^{(o)}) = \frac{1}{2}(\mathcal{L}_{j,o}^{(\mathrm{p},1)}(\Theta_j^{(o)}) + \mathcal{L}_{j,o}^{(\mathrm{p},2)}(\Theta_j^{(o)})). \tag{17}$$

Note that this loss function depends on the averaged GGD parameters $\tilde{\beta}_{j+1}^{(o)}$ and $\tilde{\alpha}_{j+1}^{(o)}$ given by (10) and (13), respectively. In this respect, and for the sake of simplicity (to avoid generating the wavelet coefficients of each image and estimating the GGD parameters of the resulting subbands for several epochs), $\tilde{\beta}_{j+1}^{(o)}$ and $\tilde{\alpha}_{j+1}^{(o)}$ are computed over the training dataset only once time before starting the learning process (i.e., without updating them after each epoch).

We will see in the next section how the defined loss function can be employed to develop a new rate-distortion strategy.

IV. Towards Joint Learning Prediction and Update FCNN Models

A. Motivation

To learn the FCNN models involved in our NSLS architecture, different approaches can be envisaged. A straightforward solution, adopted in [10], consists of independently learning the FCNN models as described in Section II-C. To overcome the aforementioned limitations of such approach, a single level optimization technique could be used. This technique aims at jointly learning the FCNN models by optimizing a loss function performed on the outputs of a given resolution level. Such a learning approach, where only the three prediction models are jointly trained, independently of the update, has been developed in [25]. However, in order to deal with the strong dependencies existing between the different wavelet subbands (intra and inter scales), it appears more judicious to learn the involved FCNN prediction and update models while taking into account all the generated coefficients. Such an approach, performed across the different resolution levels, will be referred to as a multi-scale optimization technique.

B. Multi-Scale Optimization Techniques for Joint Learning of Prediction and Update Models

To achieve this goal, our lifting-based multiresolution structure will be interpreted as a global architecture whose different FCNN prediction and update models will be jointly learned. In this respect, and instead of using different loss functions specific to the prediction and update models, these models will be learned at once through a *unique* loss function. Two expressions of the loss functions will be investigated subsequently.

1) Combined Predict-Update Loss Functions: Let Θ denote the set of weights associated with all the FCNN prediction and update models used across the J resolution levels:

$$\Theta = \left(\Theta_j^{(o)}\right)_{\substack{o\in\{HH,LH,HL,LL\}\\ j\in\{0,\ldots,J-1\}}}. \tag{18}$$

To learn the resulting vector of parameters, a first approach consists in combining the two loss functions previously used for optimizing the prediction and update models, respectively. Thus, the corresponding loss function is a weighted sum of two terms evaluated on the multiresolution representation of each training image. The first term represents the sum of the entropy of all the detail subbands while the second one aims at ensuring that the subband x_J, obtained at the resolution level J, represents a coarse approximation to the original image. Therefore, our first loss function used for jointly learning the prediction and update models is given by

$$\begin{aligned}&\mathcal{L}_1^{(\mathrm{p,u})}(\Theta)\\ &= \sum_{j=0}^{J-1}\frac{1}{4^{j+1}}\sum_{o\in\{HH,LH,HL\}}\mathcal{L}_{j,o}^{(\mathrm{p})}(\Theta_j^{(o)}) + \frac{\lambda_1}{4^J}\tilde{\mathcal{L}}^{(\mathrm{u})}(\Theta_{J-1}^{(LL)})\end{aligned} \tag{19}$$

with λ_1 is a positive constant weighting the predict and update loss functions. Note that the factor $\frac{1}{4^{j+1}}$ corresponds to the ratio between the size of the wavelet subband $x_{j+1}^{(o)}$ and that of the original image, so as to weight properly the contribution of this subband in the final image representation.

While the predict loss function $\mathcal{L}_{j,o}^{(\mathrm{p})}$ is defined in (17), the update one $\tilde{\mathcal{L}}^{(\mathrm{u})}$ will be expressed as

$$\begin{aligned}&\tilde{\mathcal{L}}^{(\mathrm{u})}(\Theta_{J-1}^{(LL)})\\ &= \sum_{m=1}^{M_{J-1}}\sum_{n=1}^{N_{J-1}}\left(y_J(m,n) - x_{0,0}(m,n) - \hat{t}_{J-1}(m,n)\right)^2\end{aligned} \tag{20}$$

where y_J is obtained by applying J successive operations of convolution to the original image x_0, using the ideal low-pass filter $\tilde{h}$ (see Eq. (6)), followed by a decimation of factor 2.

2) Rate-Distortion Approximation Based Loss Function: In the context of lossy coding application, a more appealing learning approach would consist in optimizing all the FCNN models by minimizing a Rate-Distortion (R-D) based cost function. Note that most of the existing end-to-end learning image compression methods use a R-D based loss function where the quantization is replaced with an additive uniform noise on the unit interval. This is achieved while assuming a fixed uniform scalar quantizer. However, such a common approach cannot be easily exploited in our subband-based image coding context where variable quantization steps are generally assigned to the different subbands in the test coding phase.

As a result, we propose an alternative solution based on a coarse-to-fine coding/decoding strategy. More precisely, while the rate can again be approximated by the weighted sum of the entropies of all the subbands, a both simple and efficient distortion evaluation will be performed. First, we compute the reconstruction error (MSE) between the original image and the reconstructed one from the approximation coefficients at the coarsest resolution level J, i.e by assuming that the remaining detail coefficients are set to zero. Then, similar reconstruction errors are evaluated and accumulated while progressively adding the detail subbands. Therefore, our new R-D based loss function, denoted by $\mathcal{L}_2^{(\mathrm{p,u})}$, can be defined as follows:

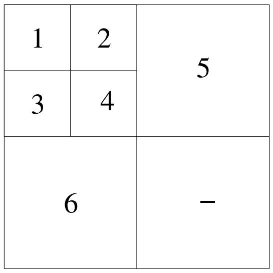


Fig. 4. The order of the retained i wavelet subbands (with $i \in \{1,\ldots,3J\}$, with $J = 2$) during reconstruction error evaluation.

$$\mathcal{L}_2^{(\mathrm{p,u})}(\Theta) = R(\Theta) + \lambda_2 D(\Theta) \tag{21}$$

where λ_2 is a positive constant weighting the rate and distortion criteria, given respectively by

$$\begin{aligned}R(\Theta) = &\sum_{j=0}^{J-1}\frac{1}{4^{j+1}}\sum_{o\in\{HH,LH,HL\}}\mathcal{L}_{j,o}^{(\mathrm{p})}(\Theta_j^{(o)})\\ &+\frac{1}{4^J}\mathcal{L}_{J-1,LL}^{(\mathrm{p})}(\Theta_{J-1}^{(LL)}),\end{aligned} \tag{22}$$

and

$$D(\Theta) = \sum_{i=1}^{3J}\mathrm{MSE}(x, \tilde{x}_i(\Theta)). \tag{23}$$

Hereabove, $\tilde{x}_i$ corresponds to the reconstructed image obtained through the synthesis stage of the FCNN structure while using the first i-th subbands of the multiresolution representation.

To evaluate (23), the subbands are added following a coarse-to-fine resolution order as shown in Fig. 4. Note that the diagonal detail subband obtained at the finest resolution level is not considered in the last synthesis stage. This is because adding the last subband will result in a reconstruction error between x and $\tilde{x}_{3J+1}$ equal to zero in theory, due to the perfect reconstruction property of the lifting scheme. The parameters $\tilde{\alpha}_{j+1}^{(o)}$ and $\tilde{\beta}_{j+1}^{(o)}$, used in the above two loss functions, have been defined in (13) and depend on the orientation of the detail subband as well as its resolution index. Moreover, whatever the employed loss function ($\mathcal{L}_1^{(\mathrm{p,u})}$ or $\mathcal{L}_2^{(\mathrm{p,u})}$), it is minimized using the MBGD algorithm. After training, the optimized weight vector Θ_j containing the learned FCNN prediction and update models is applied to the test images to generate their respective multiresolution representations (using Eqs. (2) and (3)), and they are subsequently encoded.

V. Experimental Results

In this section, the proposed joint learning of the prediction and update FCNN models in lifting based coding schemes

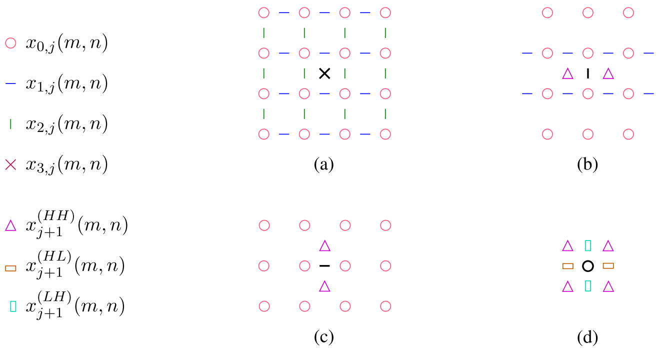


Fig. 5. Spatial supports of the prediction and update operators used to generate: (a) the diagonal detail coefficients $x_{j+1}^{(HH)}$, (b) the vertical detail coefficients $x_{j+1}^{(LH)}$, (c) the horizontal detail coefficients $x_{j+1}^{(HL)}$, and (d) the approximation coefficients x_{j+1}. Note that for every step, the pixels to be predicted and updated are highlighted in black.

will be evaluated. Our approach will be compared to different state-of-the-art neural networks based compression methods.

A. Experimental Settings

To evaluate the performance of the proposed approach, our FCNN-based lifting architectures have been trained using the Challenge on Learned Image Compression (CLIC) database.[1] The training dataset consists of 585 images of various sizes. The compression methods are then validated on the test CLIC dataset by randomly selecting 40 crop images of size 512 × 512. In addition, we have also considered 30 images, of size 1200 × 1200, taken from the Tecnick sampling dataset[2] [48], [49]. Note that all the tested methods will be evaluated on the luminance component of the CLIC and Tecnick images since our FCNN-based LS is designed to process 2D images (i.e., a single component).

Regarding the different FCNN prediction and update models, we have employed: 4 hidden layers of size 128 × 64 × 32 × 16, Parametric Rectified Linear Unit (PReLU) activation functions, and a learning rate equal to 10^{-3} while applying a decay of 10^{-4}. The simulations were carried out by using Keras and TensorFlow on an NVIDIA Tesla V100 32 GB GPU. Finally, the initialization steps of the joint learning approaches used the pre-trained FCNN models obtained with the independent learning strategy.[3]

B. Comparison Methods

First, the prediction and update steps of the proposed lifting structure have been performed using the 2D spatial supports shown in Fig. 5. Recall that these supports represent the samples assigned to the input layers of the different FCNN models. Our proposed FCNN-based LS decompositions, carried out over three resolution levels, will be designated as follows.

- FCNN-LS-JL-ML1 represents the first improved version of the FCNN-based LS method, described in Section IV-B, where *both* FCNN prediction and update models are jointly learned using a multi-level optimization technique based on the combined predict-update loss function $\mathcal{L}_1^{(\text{p,u})}$ in (19).
- FCNN-LS-JL-ML2 is the second improved version of the FCNN-based LS. It is similar to the previous one, except that we resort to the rate-distortion based loss function $\mathcal{L}_2^{(\text{p,u})}$ in (21).

Note that these two approaches are tested using four λ_1 and λ_2 values (as discussed at the beginning of Section V-D) and the best model in terms of R-D performance is selected.

The above methods will be compared to the following state-of-the-art deep learning-based coding methods:

- AE-Fact [33] is among the first reference end-to-end optimized image compression methods and relies on a nonlinear transform composed of three successive stages of linear filters (convolutions) and nonlinear activation functions. It uses a non-adaptive distribution model, based on piecewise linear functions, referred to as *factorized-prior* model.
- AE-Hyp [36] corresponds to an extension of the previous one and aims to integrate a *hyperprior* model to capture the spatial dependencies of the latent representation. To do so, a zero-mean Gaussian distribution with standard deviation parameter σ^2 is considered.
- AE-Hyp-GMM [39] is also an improved version of the previous architectures and uses a *Gaussian mixture model* (GMM) with an attention module.
- iwave [23] is a recent method more related to this work. It resorts to a neural network-based LS where the update step is a mean filter and the prediction one is performed using a CNN.
- iwave++ [24] corresponds to an extended version of the "iwave" by applying CNN to both prediction and update steps and optimizing the architecture in an end-to-end fashion. Note that the tested method corresponds to the *lossy multi-model* "iwave++", which is the only version available on GitHub.

All these NN-based coding methods (except "iwave++ [24]") have been developed for lossy image compression purposes and as such, they will be considered as benchmarks in terms of R-D performance. However, "iwave [23]" can be easily exploited as a lossless compression method.

Furthermore, the proposed methods will also be compared to our previous FCNN-based lifting schemes [10], [25], through an ablation study, to show the benefits of the developed joint multi-scale learning techniques compared to the independent and single level learning approaches. In addition to the aforementioned deep learning methods, the proposed ones will be compared to some popular image coding standards including JPEG2000 and BPG.

It should be noted that for the proposed FCNN-LS and "iwave [23]" methods, JPEG2000 has been used *only* as an entropy encoder. For instance, the analysis and synthesis stages of JPEG2000 (based on 5/3 and 9/7 transforms) have been disabled and replaced by those of the tested FCNN-LS or "iwave".

[1] http://www.compression.cc/2018/challenge/

[2] https://testimages.org/

[3] The source code of the proposed methods as well as the trained FCNN prediction and update models will be made publicly available.

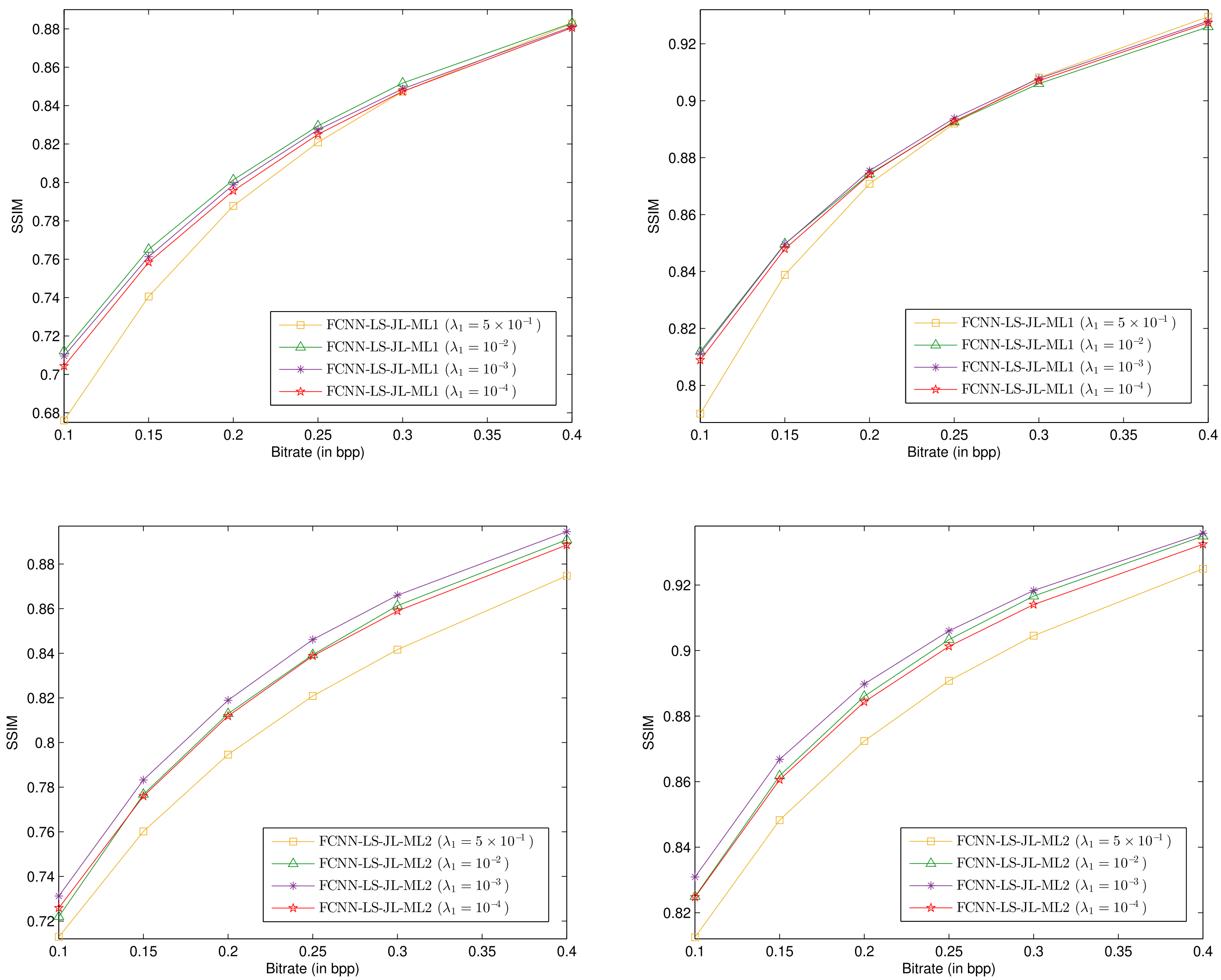


Fig. 6. Average R-D results of FCNN-LS-JL-ML1 (first row) and FCNN-LS-JL-ML2 (second row) with different λ_k values, with $k \in \{1, 2\}$, for the CLIC (first column) and Tecnick (second column) image datasets.

C. Performance Metrics

The proposed methods have been evaluated in the context of lossy as well as lossless compression using various criteria. In the context of lossless compression, we have considered the entropy of the multiresolution representation as well as the real bitrate of the encoded image. While the bitrate is measured using the JPEG2000 encoding module, the entropy value of a given wavelet representation is expressed as

$$\mathcal{H} = \sum_{j=1}^{J} \sum_{o \in \{HL,LH,HH\}} \frac{1}{4^j} \mathcal{H}_j^{(o)} + \frac{1}{4^J} \mathcal{H}_J^{(LL)} \tag{24}$$

where $\mathcal{H}_j^{(o)}$ is the entropy of the wavelet subband $x_j^{(o)}$ at resolution level j and orientation o.

The lossy compression performance are illustrated in terms of R-D results. To assess the quality of reconstructed images at different bitrates, three metrics are employed. The first and second ones are the widely used Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) metrics [50]. The third one is the Perceptual Image-Error Assessment through Pairwise Preference (PieAPP) metric [51], which was found to be better-correlated with human perception than its counterparts such as PSNR and SSIM. It is worth pointing out that such traditional metrics, and more specifically the PSNR, were found to be much less accurate to assess the visual quality of the reconstructed images in recent studies devoted to the quality assessment aspects of deep learning based compressed images [52], [53], [54]. Finally, the Bjøntegaard metric [55] has been employed to evaluate the R-D performance in terms of bitrate saving and image quality enhancement.

D. Results and Discussion

1) Impact of Parameters:

- Influence of λ_1 and λ_2 values: Our proposed FCNN-LS-JL-ML1 and FCNN-LS-JL-ML2 methods, whose loss functions are given by (19) and (21), were first evaluated using different λ_1 and λ_2 values. More precisely, for each $k \in \{1, 2\}$, we considered $\lambda_k \in \{10^{-4}, 10^{-3}, 10^{-2}, 5 \times 10^{-1}\}$. The impact of these λ_k values on the coding performance is shown in Fig. 6 and Table I in terms of R-D and entropy results, respectively.

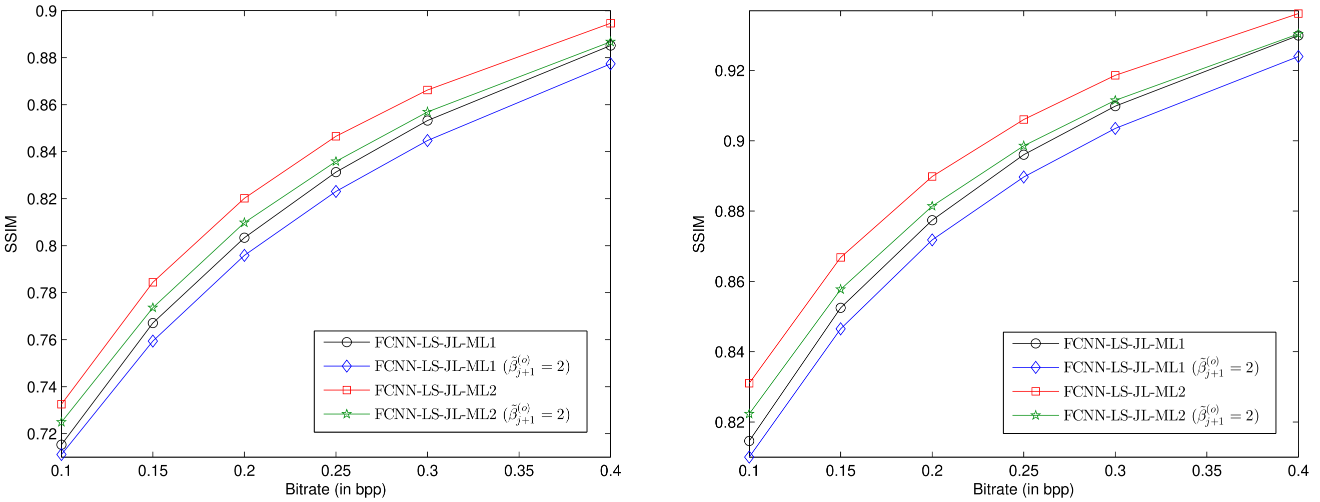


Fig. 7. Average R-D results of FCNN-LS-JL-ML1 and FCNN-LS-JL-ML2 with different $\tilde{\beta}^{(o)}_{j+1}$ values for the CLIC (first column) and Tecnick (second column) image datasets.

TABLE I

AVERAGE ENTROPY RESULTS (IN BPP) OF FCNN-LS-JL-ML1 AND FCNN-LS-JL-ML2 WITH DIFFERENT λ_k VALUES, WITH $k \in \{1, 2\}$, FOR THE CLIC AND TECNICK (SECOND COLUMN) IMAGE DATASETS

Method	CLIC	Tecnick
FCNN-LS-JL-ML1 ($\lambda_1 = 5 \times 10^{-1}$)	4.14	3.77
FCNN-LS-JL-ML1 ($\lambda_1 = 10^{-2}$)	4.10	3.74
FCNN-LS-JL-ML1 ($\lambda_1 = 10^{-3}$)	4.09	3.73
FCNN-LS-JL-ML1 ($\lambda_1 = 10^{-4}$)	4.09	3.73
FCNN-LS-JL-ML2 ($\lambda_2 = 5 \times 10^{-1}$)	4.39	3.95
FCNN-LS-JL-ML2 ($\lambda_2 = 10^{-2}$)	4.18	3.79
FCNN-LS-JL-ML2 ($\lambda_2 = 10^{-3}$)	4.12	3.75
FCNN-LS-JL-ML2 ($\lambda_2 = 10^{-4}$)	4.10	3.74

TABLE II

AVERAGE ENTROPY (IN BPP) RESULTS OF FCNN-LS-JL-ML1 AND FCNN-LS-JL-ML2 WITH DIFFERENT $\tilde{\beta}^{(o)}_{j+1}$ VALUES FOR THE CLIC AND TECNICK IMAGE DATASETS

Method	CLIC	Tecnick
FCNN-LS-JL-ML1 ($\tilde{\beta}^{(o)}_{j+1} = 2$)	4.13	3.75
FCNN-LS-JL-ML1	4.09	3.73
FCNN-LS-JL-ML2 ($\tilde{\beta}^{(o)}_{j+1} = 2$)	4.14	3.76
FCNN-LS-JL-ML2	4.10	3.74

For the first multi-scale optimization approach, the best results are obtained with $\lambda_1 = 10^{-2}$ at low bitrates and $\lambda_1 = 5 \times 10^{-1}$ at higher bitrates. However, based on the R-D plots obtained with the different images of both datasets, we observed that the appropriate choice for the λ_1 value may change from one image to another. Regarding the second-multi-scale optimization technique, it is important to note that using a single model obtained with $\lambda_2 = 10^{-3}$ leads to a good coding performance in terms of SSIM at different bitrates. However, and similar to the first approach, it has been observed with the PieAPP metric that the different test images may have various appropriate λ_2 values. For these reasons, we propose to apply the four models obtained with $\lambda_k \in \{10^{-4}, 10^{-3}, 10^{-2}, 5 \times 10^{-1}\}$ and then select the best one for each target bitrate of the test images. Note that these hybrid approaches, designated as FCNN-LS-JL-ML1 and FCNN-LS-JL-ML2 in what follows, will result in a negligible overhead (2 bits per image) that needs to be sent to the decoder.

Unlike the R-D coding performance, the results of the proposed multi-scale optimization techniques in terms of entropy, shown in Table I, indicate that setting λ_1 (or λ_2) to 10^{-4} leads to the lowest entropy values for all the dataset images. Therefore, it will be enough to employ a single model (corresponding to $\lambda_1 = \lambda_2 = 10^{-4}$) in the context of lossless compression.

- Influence of $\tilde{\beta}^{(o)}_{j+1}$ parameters: We have also studied the impact of the $\tilde{\beta}^{(o)}_{j+1}$ parameter, used in the $\ell_{\tilde{\beta}^{(o)}_{j+1}}$-norm arising in (17), on the proposed FCNN-LS-JL-ML1 and FCNN-LS-JL-ML2 methods. While the most commonly used criterion consists in setting $\tilde{\beta}^{(o)}_{j+1}$ to 2, we have proposed here to use adaptive $\tilde{\beta}^{(o)}_{j+1}$ values, provided in the captions of Fig. 3, which depend on the orientation of the detail wavelet subbands as well as their resolution levels. Table II and Fig. 7 show the coding performance of these methods in terms of entropy and R-D results. Observe that using adaptive $\tilde{\beta}^{(o)}_{j+1}$ parameters yields better coding performance, especially in the context of lossy compression.

2) Lossy Coding Performance: An objective evaluation of the aforementioned coding methods is first performed in the context of lossy compression. The average R-D results, obtained with CLIC and Tecnick datasets, are shown in Fig. 8. Note that higher SSIM values and lower PieAPP values indicate better quality of reconstruction. According to the PSNR values, JPEG2000 appears more performant than some deep learning based image coding methods. On the other hand, the SSIM plots show that most of these deep learning approaches yields better results compared to JPEG2000. For this reason, and based on the recent image quality assessment studies showing the limitations of these traditional metrics (as discussed in Section V-C), we proposed to

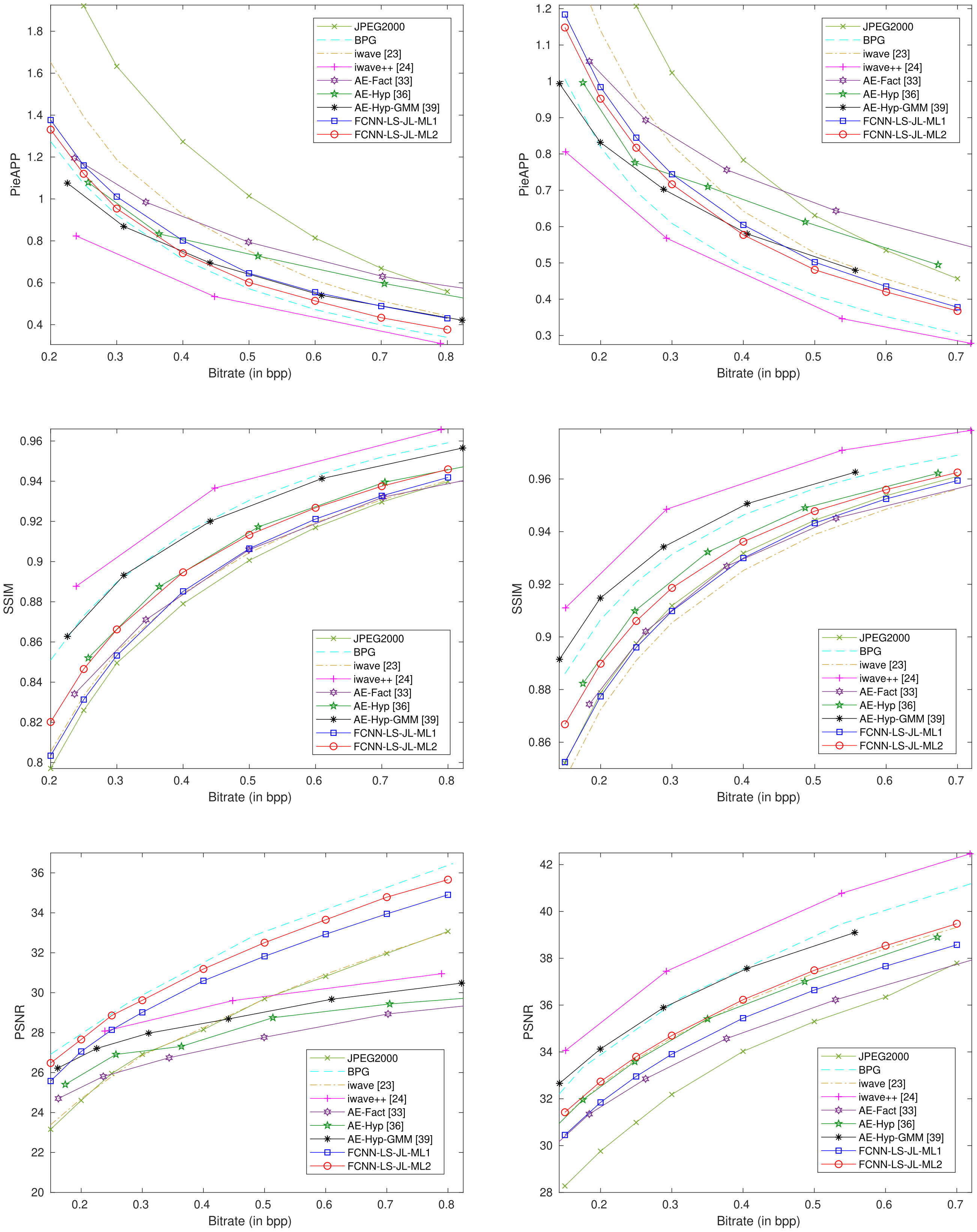


Fig. 8. Average R-D results of the CLIC (first column) and Tecnick (second column) image datasets using PieAPP, SSIM and PSNR metrics.

resort to a more recent perceptual metric (PieAPP), which was found to be more accurate to judge the visual quality of the decoded images [10]. Thus, the PieAPP metric confirms that most of deep learning based image compression methods outperforms JPEG2000. Most importantly, the two proposed multi-scale learning approaches, and in particular the

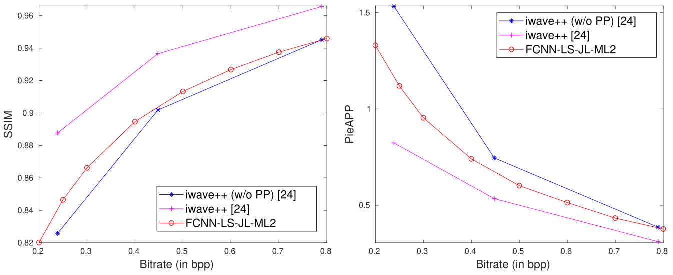


Fig. 9. Removing the post-processing (PP) module in iwave++ [24] and comparison with the proposed method: R-D results for the CLIC dataset.

TABLE III

Bjøntegaard Metric: The Average PieAPP Difference and the Bitrate Saving

(a) The gain of "FCNN-LS-JL-ML2" w.r.t "iwave [23]".

	bitrate saving (in %)			PieAPP difference		
Datasets	low	middle	high	low	middle	high
CLIC	-35.42	-25.04	-16.18	-0.42	-0.20	-0.06
Tecnick	-32.55	-14.98	-9.01	-0.29	-0.08	-0.02

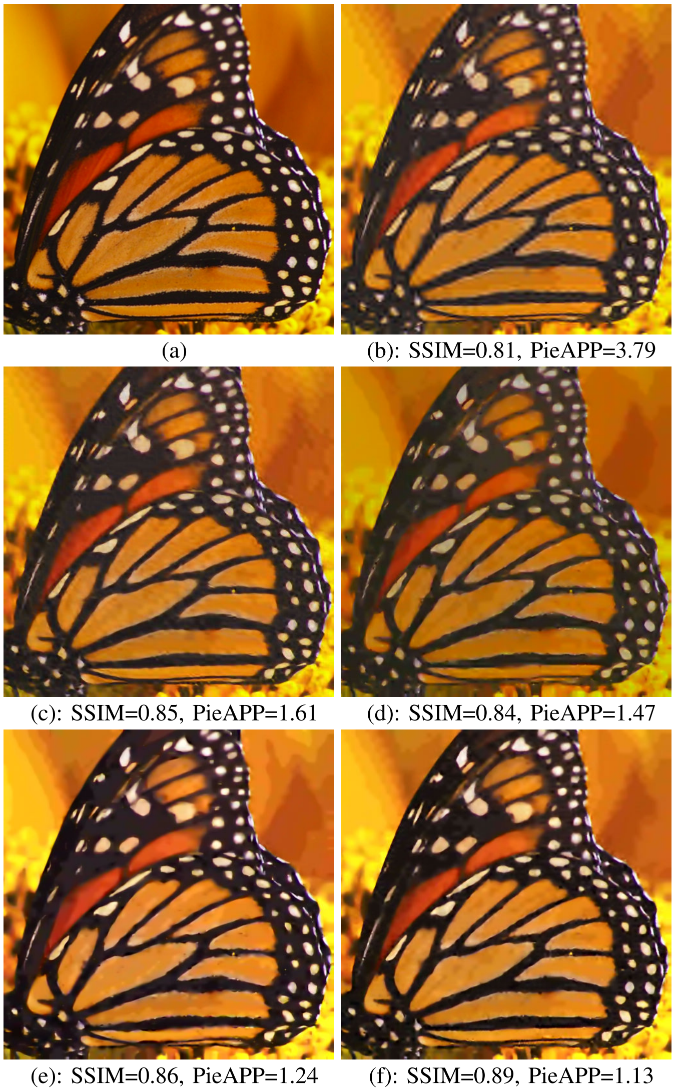
(b) The gain of "FCNN-LS-JL-ML2" w.r.t "FCNN-LS-IL".

	bitrate saving (in %)			PieAPP difference		
Datasets	low	middle	high	low	middle	high
CLIC	-5.91	-9.41	-9.50	-0.06	-0.07	-0.04
Tecnick	0.72	-3.38	-13.95	0.01	-0.02	-0.04

second (i.e., FCNN-LS-JL-ML2), lead to better results compared to different existing deep learning techniques (except iwave++ [24]). Note that the good performance obtained with iwave++ [24] is due to the integration of a post-processing model in the developed compression method. For fairer evaluation, the proposed methods should be compared to iwave [23] (rather than iwave++ [24]) since iwave only focuses on the use of neural networks to improve lifting based decomposition (as investigated in this paper). For instance, by removing the Post-Processing (PP) module in iwave++ [24], the latter (designated by "iwave++ (w/o PP) [24]") becomes less performant than the proposed method FCNN-LS-JL-ML2 as depicted in Fig. 9. Finally, our FCNN-LS-JL-ML2 method is slightly worse than BPG in terms of PieAPP metric. However, our method achieves a significant gain compared to BPG in the lossless compression context as it will be shown later.

In addition, the proposed method FCNN-LS-JL-ML2 has been compared to the closely related work "iwave [23]" using the Bjøntegaard metric. Table III.(a) shows the relative gains at low, middle and high bitrates corresponding respectively to the bitrates $\{0.07, 0.1, 0.15, 0.2\}$, $\{0.25, 0.3, 0.4, 0.5\}$ and $\{0.7, 0.8, 0.9, 1\}$ bpp. Note that a negative value in terms of bitrate saving (resp. PieAPP difference) indicates a decrease of bitrate (resp. PieAPP) for the same PieAPP (resp. bitrate). Thus, it can be seen that our proposed method achieves significant bitrate savings at different bitrates. These bitrate savings reach about 35%, 25%, and 16% at low, middle, and high bitrates.

A subjective evaluation of the decoded images at different bitrates is also conducted to confirm the good performance of the proposed FCNN-based LS coding methods. Figures 10 and 11 illustrate some reconstructed images for various methods. It can be seen that the proposed FCNN-LS-JL-ML2 based coding method leads to a better visual reconstruction quality compared to the other methods. In particular, our reconstructed images present better contrast while preserving sharp edges.



(a)

(b): SSIM=0.81, PieAPP=3.79

(c): SSIM=0.85, PieAPP=1.61

(d): SSIM=0.84, PieAPP=1.47

(e): SSIM=0.86, PieAPP=1.24

(f): SSIM=0.89, PieAPP=1.13

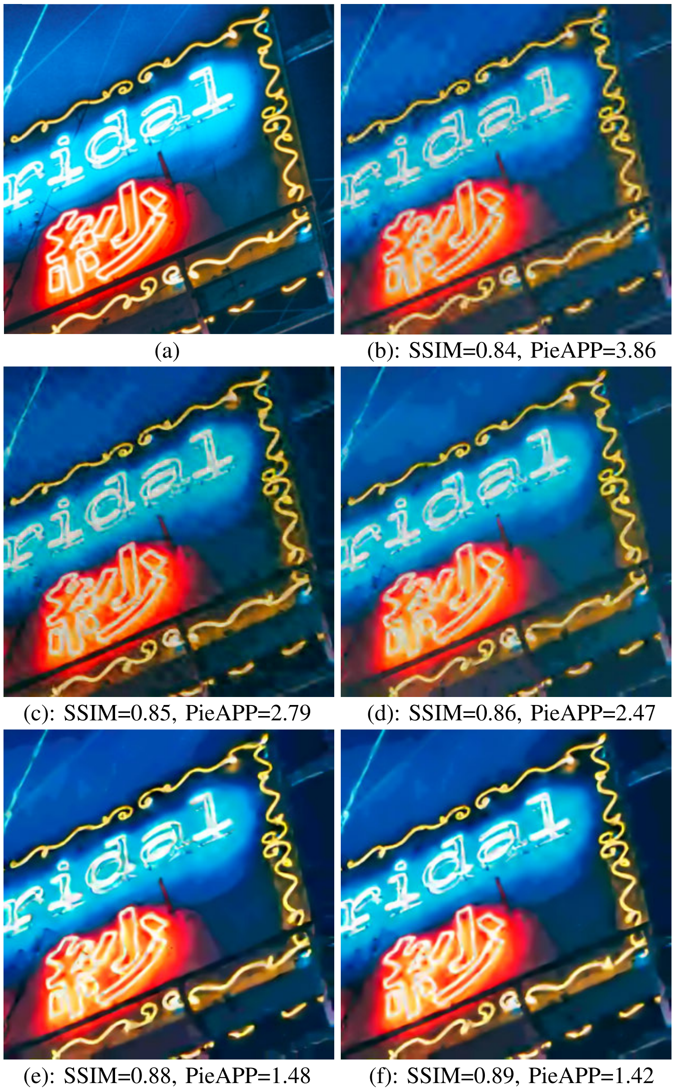
Fig. 10. (a) Original test image (image 21 of the CLIC dataset). The reconstructed ones at 0.1 bpp using: (b) JPEG2000, (c) CNN-LS [23], (d) FCNN-LS-JL-SL [25], (e) FCNN-LS-JL-ML1, (f) FCNN-LS-JL-ML2.

TABLE IV

Lossless Compression Performance in Terms of Average Bitrate (in bpp)

Method	CLIC	Tecnick	Kodak
JPEG2000	4.16	3.72	4.67
BPG	4.09	4.03	4.73
iwave [23]	4.66	4.17	4.99
FCNN-LS-IL	4.09	3.64	4.58
FCNN-LS-JL-SL	4.07	3.62	4.56
FCNN-LS-JL-ML1	4.00	3.58	4.49
FCNN-LS-JL-ML2	4.00	3.58	4.49

3) Lossless Coding Performance: Since our FCNN-LS based methods are applicable to lossy-to-lossless coding, they have also been evaluated in the context of lossless compression. Table IV provides the average bitrate values of different



(a) (b): SSIM=0.84, PieAPP=3.86

(c): SSIM=0.85, PieAPP=2.79 (d): SSIM=0.86, PieAPP=2.47

(e): SSIM=0.88, PieAPP=1.48 (f): SSIM=0.89, PieAPP=1.42

Fig. 11. (a) Original test image (image 3 of the CLIC dataset). The reconstructed ones at 0.2 bpp using: (b) JPEG2000, (c) CNN-LS [23], (d) FCNN-LS-JL-SL [25], (e) FCNN-LS-JL-ML1, (f) FCNN-LS-JL-ML2.

lossless coding methods for the CLIC, Tecnick and Kodak image datasets. It can be first observed that the modified version of iwave [23] is less performant than JPEG2000 and so, it is not efficient for lossless image coding. However, our proposed joint learning approaches significantly outperform iwave [23]. Moreover, they achieve a gain of about 0.15 bpp compared to the JPEG2000 coding standard. Finally, our methods are more performant than BPG yielding a gain of about 0.1 bpp, 0.45 bpp and 0.25 bpp on CLIC, Tecnick and Kodak datasets, respectively. It should be emphasized that the two variants of the proposed multi-scale optimization technique (FCNN-LS-JL-ML1 and FCNN-LS-JL-ML2) lead to similar lossless coding performance. Note that the comparison with other variants of FCNN-based lifting schemes will be later discussed in the ablation study.

4) Ablation Study: This study aims to illustrate the role of the different loss functions used to learn the involved prediction and update models in our FCNN-based lifting architecture. In this respect, the proposed joint learning techniques

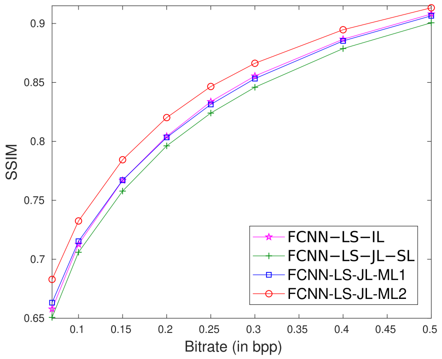


Fig. 12. R-D results for the CLIC dataset using different learning techniques of the proposed FCNN based lifting architecture.

FCNN-LS-JL-ML1 and FCNN-LS-JL-ML2, based on the loss functions $\mathcal{L}_1^{(\mathrm{p,u})}$ (19) and $\mathcal{L}_2^{(\mathrm{p,u})}$ (21), have been compared to the two following approaches:

- FCNN-LS-IL where an *Independent Learning* approach [10] is applied using the prediction and approximation errors based loss functions $\widetilde{\mathcal{L}}^{(\mathrm{p})}$ and $\widetilde{\mathcal{L}}^{(\mathrm{u})}$ defined in (4) and (8), respectively.
- FCNN-LS-JL-SL where the FCNN prediction models are *jointly* learned using a *single level* optimization technique [25] (i.e., by minimizing a weighted sum of the loss functions $\widetilde{\mathcal{L}}^{(\mathrm{p})}$ evaluated on a given resolution level).

For fair comparison, since the proposed joint learning techniques have been designed using adaptive $\tilde{\beta}_{j+1}^{(o)}$ values, the independent and single level optimization techniques have also been evaluated using the same $\tilde{\beta}_{j+1}^{(o)}$ values instead of setting them to 2 as performed in [10] and [25]. Let us recall that the benefits of using adaptive $\tilde{\beta}_{j+1}^{(o)}$ parameters has been shown at the beginning of Section V-D and Fig. 7.

Fig. 12 illustrates the R-D performance of the different learning strategies. It can be noticed that the single level learning approach is slightly less performant than the independent strategy, which suggests that the joint optimization of *only* the prediction models (per resolution level) is still suboptimal. Thus, by resorting to a multi-level optimization technique and combining the prediction and update loss functions, the FCNN-LS-JL-ML1 improves the R-D results. Finally, further improvements are achieved using the R-D approximation based loss function (i.e., FCNN-LS-JL-ML2). The latter has also been compared to the independent learning approach FCNN-LS-IL using the Bjøntegaard metric. Table III.(b) shows the relative gains at low, middle, and high bitrates (using the same bitrates defined with Table III.(a)). The proposed method outperforms the independent learning approach and reaches a bitrate saving of about 6%, 9%, and 14% at low, middle, and high bitrates.

Moreover, in the context of lossless compression, Table IV shows that the single level optimization technique (i.e "FCNN-LS-JL-SL") results in a slight improvement of 0.02 bpp compared to the independent learning approach (i.e "FCNN-LS-IL"). This gain reaches 0.1 bpp by applying the multi-level optimization technique.

5) Computational Complexity Analysis: Finally, the proposed method is evaluated and compared to the recent related

TABLE V
COMPLEXITY OF THE PROPOSED METHOD

Criterion	iwave++ [24]	FCNN-LS-JL-ML2
Number of parameters	17.91 M	167244
FLOPs	5595.4 G	43.2 G
Encoding time	22.3 s	8.5 s
Decoding time	8.65 s	3.2 s

work “iwave++ [24]” in terms of encoding/decoding time, number of model parameters, and number of floating point operations per second (FLOPs). Table V illustrates this comparison for an image of size 1200 × 1200 using an Intel Xeon(R) processor (4 GHz) and a Python implementation. It can be observed that “iwave++ [24]” employs a large amount of parameters (17.91 M) and FLOPs (5595 G). However, the proposed FCNN-LS-JL-ML2 method uses only 167244 trainable parameters and 43,2 GFLOPs. Moreover, the proposed method requires 8.5/3.2 seconds for the encoding/decoding process, which is about 2.7 times faster than “iwave++ [24]”. The runtime of our method is reduced to 0.3/0.08 seconds when the code is executed on an NVIDIA Tesla V100 32Gb GPU. The main difference in the computational complexity between the two architectures is explained by the fact that “iwave++ [24]” uses neural networks for three main modules including lifting based decomposition, entropy coding, and post-processing, while our method only focuses on the first module. It should be noted here that the computational complexity has been only given for FCNN-LS-JL-ML2 in Table V since it is the same for all the proposed FCNN-LS. Indeed, the different FCNN-LS schemes use the same architecture and the main difference between them only concerns the learning approach of the FCNN prediction and update models. While the different end-to-end learning methods often use a specific model for each point of the R-D curve which will yield multiple models covering low, middle and high bitrates, it is worth pointing out that the proposed method has the additional main advantage of learning a single model that can be used at different bitrates as shown at the beginning of Section V-D and Fig. 6. Therefore, the proposed method corresponds to a light model and allows a fast encoding/decoding process while being suitable for lossy-to-lossless compression.

All these results confirm the effectiveness of the proposed FCNN design strategies, and the flexibility offered by the joint learning approaches.

VI. CONCLUSION AND PERSPECTIVES

Novel learning approaches for the design of FCNN based lifting coding schemes have been proposed in this paper. While a straightforward approach consists in separately learning all the involved FCNN models, we have investigated the benefits of joint learning approaches. These approaches allow us to take into account the dependencies existing between the different models as well as the multiresolution aspect of the LS architecture. In doing so, an entropy based loss function and a multi-scale optimization technique have been developed. Experimental results have shown the effectiveness of the proposed approaches in the context of both lossy and lossless compression. In future work, an extension of the proposed FCNN-LS architecture to more sophisticated structures, like vector lifting scheme [56], could be envisaged to deal with multi-component images. Another road of investigation could be an end-to-end learning approach taking into account the quantization and entropy coding modules.

REFERENCES

[1] B. Pesquet-Popescu and V. Bottreau, “Three-dimensional lifting schemes for motion compensated video compression,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 3, Salt Lake City, Utah, May 2001, pp. 1793–1796.

[2] T. Guo, H. S. Mousavi, T. H. Vu, and V. Monga, “Deep wavelet prediction for image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1100–1109.

[3] D. Taubman, “High performance scalable image compression with EBCOT,” *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[4] M. Kaaniche, A. Benazza-Benyahia, B. Pesquet-Popescu, and J.-C. Pesquet, “Vector lifting schemes for stereo image coding,” *IEEE Trans. Image Process.*, vol. 18, no. 11, pp. 2463–2475, Nov. 2009.

[5] Y. Xing, M. Kaaniche, B. Pesquet-Popescu, and F. Dufaux, “Adaptive nonseparable vector lifting scheme for digital holographic data compression,” *Appl. Opt.*, vol. 54, no. 1, pp. 98–109, Jan. 2015.

[6] E. Martínez-Enríquez, J. Cid-Sueiro, F. Díaz-de-María, and A. Ortega, “Directional transforms for video coding based on lifting on graphs,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 4, pp. 933–946, Apr. 2018.

[7] J.-H. Jacobsen, A. W. M. Smeulders, and E. Oyallon, “I-RevNet: Deep invertible networks,” in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, May 2018, pp. 1–11.

[8] J.-J. Huang and P. L. Dragotti, “LINN: Lifting inspired invertible neural network for image denoising,” in *Proc. 29th Eur. Signal Process. Conf. (EUSIPCO)*, Dublin, Ireland, Aug. 2021, pp. 636–640.

[9] T. Dardouri, M. Kaaniche, A. Benazza-Benyahia, and J.-C. Pesquet, “Optimized lifting scheme based on a dynamical fully connected network for image coding,” in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 3329–3333.

[10] T. Dardouri, M. Kaaniche, A. Benazza-Benyahia, and J.-C. Pesquet, “Dynamic neural network for lossy-to-lossless image coding,” *IEEE Trans. Image Process.*, vol. 31, pp. 569–584, 2022.

[11] W. Sweldens, “The lifting scheme: A custom-design construction of biorthogonal wavelets,” *Appl. Comput. Harmon. Anal.*, vol. 3, no. 2, pp. 186–200, Apr. 1996.

[12] I. Daubechies and W. Sweldens, “Factoring wavelet transforms into lifting steps,” *J. Fourier Anal. Appl.*, vol. 4, no. 3, pp. 247–269, May 1998.

[13] Y.-K. Sun, “A two-dimensional lifting scheme of integer wavelet transform for lossless image compression,” in *Proc. Int. Conf. Image Process.*, vol. 1, Singapore, Oct. 2004, pp. 497–500.

[14] M. Kaaniche, A. Benazza-Benyahia, and B. Pesquet-Popescu, “Two-dimensional non separable adaptive lifting scheme for still and stereo image coding,” in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Mar. 2010, pp. 1298–1301.

[15] F. J. Hampson and J.-C. Pesquet, “M-band nonlinear subband decompositions with perfect reconstruction,” *IEEE Trans. Image Process.*, vol. 7, no. 11, pp. 1547–1560, Nov. 1998.

[16] J. Sole and P. Salembier, “Generalized lifting prediction optimization applied to lossless image compression,” *IEEE Signal Process. Lett.*, vol. 14, no. 10, pp. 695–698, Oct. 2007.

[17] Y. Liu and K. Ngi Ngan, “Weighted adaptive lifting-based wavelet transform for image coding,” *IEEE Trans. Image Process.*, vol. 17, no. 4, pp. 500–511, Apr. 2008.

[18] A. Gouze, M. Antonini, M. Barlaud, and B. Macq, “Design of signal-adapted multidimensional lifting scheme for lossy coding,” *IEEE Trans. Image Process.*, vol. 13, no. 12, pp. 1589–1603, Dec. 2004.

[19] M. Kaaniche, B. Pesquet-Popescu, A. Benazza-Benyahia, and J.-C. Pesquet, “Adaptive lifting scheme with sparse criteria for image coding,” *EURASIP J. Adv. Signal Process.*, vol. 2012, no. 1, pp. 1–22, Jan. 2012.

[20] A. Benazza-Benyahia, J.-C. Pesquet, J. Hattay, and H. Masmoudi, "Block-based adaptive vector lifting schemes for multichannel image coding," *EURASIP J. Image Video Process.*, vol. 2007, pp. 1–10, Jan. 2007.

[21] B. Pesquet-Popescu, "Two-stage adaptive filter bank," European Patent Number EP1 119 911, Jul. 27, 1999.

[22] M. Kaaniche, A. Benazza-Benyahia, B. Pesquet-Popescu, and J.-C. Pesquet, "Non-separable lifting scheme with adaptive update step for still and stereo image coding," *Signal Process.*, vol. 91, no. 12, pp. 2767–2782, Dec. 2011.

[23] H. Ma, D. Liu, R. Xiong, and F. Wu, "IWave: CNN-based wavelet-like transform for image compression," *IEEE Trans. Multimedia*, vol. 22, no. 7, pp. 1667–1679, Jul. 2020.

[24] H. Ma, D. Liu, N. Yan, H. Li, and F. Wu, "End-to-end optimized versatile image compression with wavelet-like transform," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 3, pp. 1247–1263, Mar. 2022.

[25] T. Dardouri, M. Kaaniche, A. Benazza-Benyahia, J.-C. Pesquet, and G. Dauphin, "A neural network approach for joint optimization of predictors in lifting-based image coders," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Anchorage, Alaska, Sep. 2021, pp. 3747–3751.

[26] D. Liu, H. Ma, Z. Xiong, and F. Wu, "CNN-based DCT-like transform for image compression," in *Proc. Int. Conf. MultiMedia Modeling (MMM)*, Bangkok, Thailand, Jan. 2018, pp. 61–72.

[27] E. Ahanonu, M. Marcellin, and A. Bilgin, "Lossless image compression using reversible integer wavelet transforms and convolutional neural networks," in *Proc. Data Compress. Conf.*, Snowbird, UT, USA, Mar. 2018, p. 395.

[28] P. Akyazi and T. Ebrahimi, "Learning-based image compression using convolutional autoencoder and wavelet decomposition," in *Proc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Long Beach, CA, USA, Jun. 2019, pp. 1–5.

[29] J. Li, B. Li, J. Xu, R. Xiong, and W. Gao, "Fully connected network-based intra prediction for image coding," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3236–3247, Jul. 2018.

[30] T. Dumas, A. Roumy, and C. Guillemot, "Context-adaptive neural network-based prediction for image compression," *IEEE Trans. Image Process.*, vol. 29, pp. 679–693, 2020.

[31] M. A. Yilmaz and A. Murat Tekalp, "Effect of architectures and training methods on the performance of learned video frame prediction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 4210–4214.

[32] G. Toderici et al., "Full resolution image compression with recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jul. 2017, pp. 5435–5443.

[33] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. ICLR*, Toulon, France, Aug. 2017, pp. 1–27.

[34] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proc. Int. Conf. Mach. Learn.*, Sydney, NSW, Australia, Aug. 2017, pp. 1–9.

[35] M. Li, W. Zuo, S. Gu, J. You, and D. Zhang, "Learning content-weighted deep image compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 10, pp. 3446–3461, Oct. 2021.

[36] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent.*, Vancouver, BC, Canada, May 2018, pp. 1–47.

[37] J.-H. Kim, B. Heo, and J.-S. Lee, "Joint global and local hierarchical priors for learned image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Montreal, QC, Canada, Dec. 2022, pp. 10794–10803.

[38] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative adversarial networks for extreme learned image compression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, New Orleans, LA, USA, Oct. 2019, pp. 221–231.

[39] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7936–7945.

[40] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, "Checkerboard context model for efficient learned image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 14766–14775.

[41] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning end-to-end lossy image compression: A benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4194–4211, Aug. 2022.

[42] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Practical full resolution learned lossless image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 10621–10630.

[43] F. Mentzer, L. Van Gool, and M. Tschannen, "Learning better lossless compression using lossy compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 6637–6646.

[44] I. Schiopu and A. Munteanu, "A study of prediction methods based on machine learning techniques for lossless image coding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 3324–3328.

[45] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.

[46] H. Gish and J. Pierce, "Asymptotically efficient quantizing," *IEEE Trans. Inf. Theory*, vol. IT-14, no. 5, pp. 676–683, Sep. 1968.

[47] M. N. Do and M. Vetterli, "Wavelet-based texture retrieval using generalized Gaussian density and Kullback–Leibler distance," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 146–158, Feb. 2002.

[48] N. Asuni and A. Giachetti, "Test images: A large-scale archive for testing visual devices and basic image processing algorithms," in *Proc. Eurograph. Italian Conf.*, Cagliari, Italy, Sep. 2014, pp. 1–3.

[49] N. Asuni and A. Giachetti, "TESTIMAGES: A large data archive for display and algorithm testing," *J. Graph. Tools*, vol. 17, no. 4, pp. 113–125, Oct. 2013.

[50] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[51] E. Prashnani, H. Cai, Y. Mostofi, and P. Sen, "PieAPP: Perceptual image-error assessment through pairwise preference," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, Jun. 2018, pp. 1808–1817.

[52] G. Valenzise, A. Purica, V. Hulusic, and M. Cagnazzo, "Quality assessment of deep learning-based image compression," in *Proc. Int. Workshop Multimedia Signal Process.*, Vancouver, BC, Canada, Aug. 2018, pp. 1–6.

[53] Z. Cheng, P. Akyazi, H. Sun, J. Katto, and T. Ebrahimi, "Perceptual quality study on deep learning based image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 719–723.

[54] Z. A. Khan, T. Dardouri, M. Kaaniche, and G. Dauphin, "NNCD-IQA: A new neural networks based compressed database for image quality assessment," *Multimedia Tools Appl.*, vol. 82, no. 9, pp. 13951–13971, Apr. 2023.

[55] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," ITU SG16 VCEG-M33, Austin, TX, USA, Tech. Rep., 2001.

[56] O. Dhifallah, M. Kaaniche, and A. Benazza-Benyahia, "Efficient joint multiscale decomposition for color stereo image coding," in *Proc. 22nd Eur. Signal Process. Conf. (EUSIPCO)*, Lisbon, Portugal, Sep. 2014, pp. 536–540.