Multi-Task Convolution Neural Network-Based Lifting Scheme for Image Compression

Tassnim Dardouri, Mounir Kaaniche, Amel Benazza-Benyahia, Gabriel Dauphin

Pattern Recognition Letters
journal homepage: www.elsevier.com

Multi-Task Convolution Neural Network-Based Lifting Scheme for Image Compression

Tassnim Dardouri[a], Mounir Kaaniche[b,**], Amel Benazza-Benyahia[c], Gabriel Dauphin[b]

[a]*Novelis, R & D laboratory, 75012, Paris, France*
[b]*Université Sorbonne Paris Nord, L2TI, UR 3043, F-93430, Villetaneuse, France.*
[c]*University of Carthage SUP'COM, LR11TIC01, COSIM Lab., 2083, El Ghazala, Tunisia.*

Article history:
Received xx April 2023
Received in final form xx
Accepted xx
Available online xx

Keywords: Wavelet transforms, adaptive lifting, neural networks, multi-task learning, optimization.

ABSTRACT

Lifting schemes have attracted much interest in different image processing tasks, and more specifically in the image compression field. In this context, the optimization of the lifting operators (i.e. the prediction and update ones) plays a crucial role in the design of efficient lifting-based image coding systems. In this respect, we propose in this paper to further investigate the exploitation of neural networks in a standard non-separable lifting scheme structure. More precisely, unlike previous works, where different neural network models are employed for all the prediction and update steps involved in a lifting scheme-based decomposition, our design consists in building a new multi-task convolutional neural network model that takes into account the similarities between two prediction stages. Simulations carried out on three popular image datasets show the benefits of the proposed learning-based image coding approach.

1. Introduction

Wavelets have attracted much interest in the signal/image processing community since they enable good scalability properties in quality, resolution, and rate. For instance, they have been widely adopted in various processing tasks [1, 2, 3] while handling different types of multimedia content such 2D and 3D images, video, audio, etc [4, 5]. To produce the wavelet coefficients of a given signal, Lifting Scheme (LS) was found to be an efficient tool allowing fast implementation and perfect reconstruction [6, 7]. Due to the aforementioned advantages, the LS concept has been adopted in data compression standards such as MPEG and JPEG2000 [8].

A conventional LS consists of prediction and update stages aiming at producing a set of high and low-frequency coefficients referred to as detail and approximation wavelet subbands, respectively. While the JPEG2000 image coding standard uses some predefined filters with fixed weights, many efforts have been deployed to make these weights better adapted to the input data contents and increase the coding efficiency of LS-based coders. In this respect, different optimization techniques have been developed for the design of the prediction and update operators. Most of these techniques have been devoted to the prediction filter, which is often optimized by minimizing a given criterion defined on the detail coefficients. The employed criteria include the ℓ_2 [9], ℓ_1 [10] and entropy [11, 12] measures. However, the optimization of the update operator is more challenging, and only two main techniques have been investigated in the literature. The first optimization approach aims to minimize the reconstruction error after computing the reconstructed samples from only the approximation coefficients [9, 13]. To reduce the complexity of this minimization technique, a second approach based on a simple and efficient criterion has been proposed in [14]. It consists in minimizing the error between the generated approximation subband and the target version obtained by applying an ideal low-pass filter to the input image followed by a downsampling operation. In addition to these traditional design approaches, some Neural Networks (NN)-based methods have been recently proposed. In fact, the prediction and update tasks have been performed using Convolution Neu-

**Corresponding author.
e-mail: tdardouri@novelis.io (T. Dardouri), mounir.kaaniche@univ-paris13.fr (M. Kaaniche), benazza.amel@supcom.rnu.tn (A. Benazza-Benyahia),gabriel.dauphin@univ-paris13.fr (Gabriel Dauphin)

ral Network (CNN) [15, 16] and Fully Connected Neural Network (FCNN) [17, 18, 19]. The latter, which are closely related to the current work, will be further described in Section 2.

While NN-based LS can be seen as the first category of the developed deep learning-based image compression methods, another category of methods, inspired by the auto-encoders, has also been developed in the literature. The common architecture behind most of these methods incorporates three modules, namely nonlinear analysis transform, quantization and entropy coding, and nonlinear synthesis transform [20, 21, 22]. The main differences between the aforementioned methods concern the NN models used in the analysis and synthesis transforms, and/or the employed loss function. Furthermore, NN-based intra-prediction coding techniques have also been investigated using FCNN [23] and CNN [24]. Finally, other research efforts have focused on the use of neural networks for entropy modeling in a rate-distortion optimization framework [25]. It should be noted here that most of the existing methods belong to the class of lossy compression techniques, and only a few works have been proposed for lossless compression [26, 27].

Motivated by the several advantages of lifting-based representations and the promising results shown by our recent FCNN-LS-based coding method [18], the objective of this paper is to investigate further the use of neural networks in lifting-based image coding systems. While considering a popular non-separable lifting structure that relies on three prediction stages and an update stage, we propose to perform the different involved lifting steps by using CNN models to better capture the local structure of the input image. Most importantly, unlike previous works where different neural network models are employed to carry out the LS-based decomposition at a given resolution level, a new multi-task CNN architecture is developed. The proposed architecture aims to exploit the similarities between the second and third prediction steps and perform their learning in a joint manner.

The rest of this paper is structured as follows. Section 2 presents the related works using NN-based lifting coding schemes. The proposed multi-task CNN-based LS architecture as well as the learning approaches are then described in Section 3. Finally, Section 4 illustrates the experimental results and Section 5 provides conclusions and perspectives.

2. Related works

While neural networks and lifting schemes have been recently exploited for different image processing tasks such as classification [1] and restoration [3], this section will focus on the recent NN-based LS developed for image coding purposes [15, 16, 18, 19].

In [15], 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 has been extended in [16] by applying CNN to both prediction and update stages and optimizing the architecture in an end-to-end fashion using a rate-distortion-based loss function. However, the latter suffers from two main drawbacks. First, it relies on the concept of a one-dimensional (1D) LS-based decomposition, which will increase the number of the employed NN models (and hence the number of parameters) in the whole multiresolution architecture. Moreover, the end-to-end learning strategy uses a rate-distortion-based optimization approach for different Lagrangian parameters (i.e., bitrates). Such a strategy results in multiple NN models covering a wide range of target bitrates.

To alleviate these shortcomings, we have proposed in [18] to focus on a non-separable lifting structure (NSLS) composed of three prediction stages, and an update one [10]. More precisely, let $\mathbf{X}_0$ (resp. $\mathbf{X}_j$) denote the original image (resp. the approximation subband at resolution level j). First, a split step is applied to obtain four matrices given by $\mathbf{X}_{0,j}(m,n) = \mathbf{X}_j(2m, 2n)$, $\mathbf{X}_{1,j}(m,n) = \mathbf{X}_j(2m, 2n+1)$, $\mathbf{X}_{2,j}(m,n) = \mathbf{X}_j(2m+1, 2n)$, and $\mathbf{X}_{3,j}(m,n) = \mathbf{X}_j(2m+1, 2n+1)$. Then, the prediction and update lifting stages are performed using four FCNN modules, designated by $f_j^{(o)}$ with $o \in \{HH, LH, HL, LL\}$, to produce three detail subbands oriented diagonally $\mathbf{X}_{j+1}^{(HH)}$, vertically $\mathbf{X}_{j+1}^{(LH)}$, horizontally $\mathbf{X}_{j+1}^{(HL)}$, and the approximation subband $\mathbf{X}_{j+1}$, respectively. The analysis structure of the FCNN-based NSLS architecture is shown in Fig. 1.

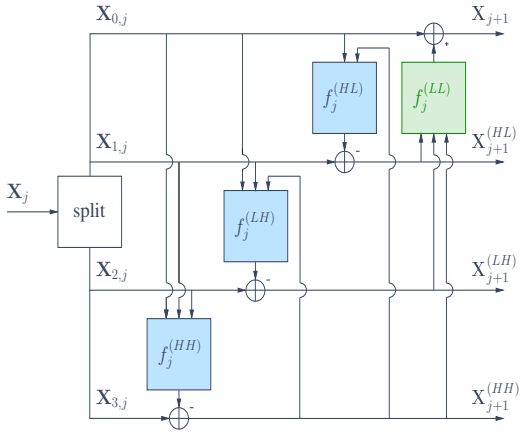


Fig. 1. Analysis structure of the FCNN-based NSLS architecture [18].

While the different FCNN models have been separately trained in [18] by minimizing the ℓ_2-norm of the prediction error, the joint learning of the three FCNN prediction models has been investigated in [19]. To this end, a weighted sum of the mean square errors has been used as a loss function. However, each of the different FCNN-based prediction and update stages has its own NN model.

3. Multi-Task CNN-based Lifting Scheme

3.1. Motivation

One main limitation of the employed FCNN model is that it does not take into account the strong local correlations in the input image. To overcome this issue and further improve the prediction performance, we first propose to resort to a CNN model due to its benefits with respect to an FCNN model [28]. For instance, in the context of intra-block prediction, it has been recently shown in [24] that CNN is more efficient than FCNN

for large blocks (of size greater than 8×8).
Moreover, in our recent works [18, 19], four different models $f_j^{(o)}$ are used to generate the approximation subband as well as the three detail subbands. However, it can be seen from Fig. 1 that, once the diagonal detail coefficients $\mathbf{X}_{j+1}^{(HH)}$ are generated, the second and third prediction steps can be performed simultaneously to produce the vertical $\mathbf{X}_{j+1}^{(LH)}$ and the horizontal $\mathbf{X}_{j+1}^{(HL)}$ detail coefficients. Furthermore, the second and third prediction steps are quite similar and share some inputs. Therefore, it becomes more interesting to design a new Multi-Task CNN (MT-CNN) model to achieve the aforementioned two prediction steps jointly.

3.2. Proposed architecture and learning approaches

The analysis structure of the proposed multi-task CNN-based NSLS architecture is depicted in Fig. 2. It consists of three CNN models. The first model, designated by $C_j^{(HH)}$, corresponds to the first prediction step that aims to generate the diagonal detail coefficients $\mathbf{X}_{j+1}^{(HH)}$. The second one, denoted by $C_j^{(HL,LH)}$, performs simultaneously the two remaining prediction tasks, to generate the vertical $\mathbf{X}_{j+1}^{(LH)}$ and the horizontal $\mathbf{X}_{j+1}^{(HL)}$ detail coefficients. Finally, the last model, designated by $C_j^{(LL)}$, will perform the update stage to produce the approximation coefficients $\mathbf{X}_{j+1}$.

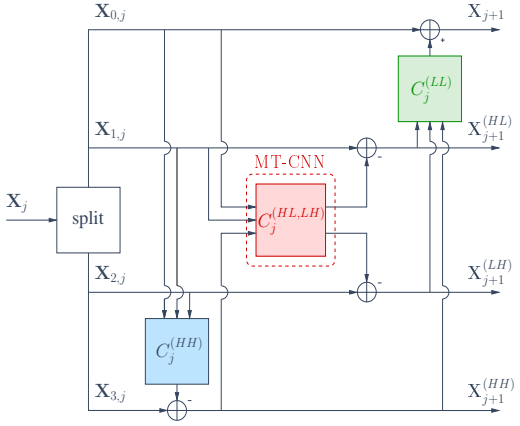


Fig. 2. Analysis structure of the proposed multi-task CNN-based NSLS.

The involved CNN models and their learning strategies will be described in what follows.

CNN-based diagonal prediction stage

The first CNN-based prediction stage aims to compute the diagonal detail coefficients as follows:

$$\begin{aligned}\mathbf{X}_{j+1}^{(HH)} &= \mathbf{X}_{3,j} - \widehat{\mathbf{X}}_{3,j} \\ &= \mathbf{X}_{3,j} - C_j^{(HH)}\left(\widetilde{\mathbf{X}}_j^{(HH)}\right)\end{aligned} \tag{1}$$

where $\mathbf{X}_{3,j}$ corresponds to the polyphase components to be predicted, and $\widehat{\mathbf{X}}_{3,j}$ represents the predicted ones obtained from the remaining components $\widetilde{\mathbf{X}}_j^{(HH)} = (\mathbf{X}_{0,j}, \mathbf{X}_{1,j}, \mathbf{X}_{2,j})$.
Thus, $\widehat{\mathbf{X}}_{3,j}$ can be viewed as the output channel of the first CNN model $C_j^{(HH)}$ whose inputs are composed of three channels $\mathbf{X}_{0,j}$, $\mathbf{X}_{1,j}$ and $\mathbf{X}_{2,j}$. The structure of the retained CNN architecture is illustrated in Fig. 3. It consists of five convolution layers using 32, 16, 16, 32, and 1 kernels, respectively. The first layer's kernel size is 7×7, whereas the remaining ones are 3×3. We also consider the Gaussian Error Linear Unit (GELU) as an activation function. It is worth pointing out that this structure has been selected based on extensive experiments taking into account the effect of several parameters (e.g., kernel size, number of layers, number of output channels, skip connection, etc) on the prediction performance.

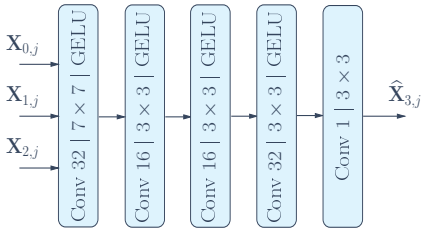


Fig. 3. Structure of the first CNN prediction model $C_j^{(HH)}$.

The retained CNN model depends on a vector of parameters $\mathbf{\Theta}_j^{(HH)}$, which is learned by minimizing the Mean Square Error (MSE) criterion. Thus, the loss function associated to the first prediction stage is given by

$$\mathcal{L}_3(\mathbf{\Theta}_j^{(HH)}) = \frac{1}{M_j N_j} \sum_{m=1}^{M_j} \sum_{n=1}^{N_j} \left(\mathbf{X}_{3,j}(m,n) - \widehat{\mathbf{X}}_{3,j}(m,n)\right)^2 \tag{2}$$

where M_j and N_j are the dimensions of the input subband $\mathbf{X}_j$ divided by 2.
Finally, the learned model is applied to each input image of the training and test datasets to predict $\mathbf{X}_{3,j}$ from $\mathbf{X}_{0,j}$, $\mathbf{X}_{1,j}$, and $\mathbf{X}_{2,j}$, and then generate the diagonal detail subband $\mathbf{X}_{j+1}^{(HH)}$ using (1).

MT-CNN-based horizontal and vertical prediction stages

Once the diagonal detail coefficients are generated, one can proceed with the second and third prediction steps to produce the vertical and horizontal detail coefficients simultaneously. Because of the similarity between these two steps, a new multi-task CNN model is proposed for the second and third prediction stages. The proposed model, based on the hard-parameter sharing scheme [29], is depicted in Fig. 4.
More precisely, the MT-CNN model consists of a shared CNN model that branches out into two task-specific models. In fact, in a typical NSLS structure (as shown in Fig. 1), the computation of the horizontal and vertical detail coefficients requires two common reference signals $\mathbf{X}_{0,j}$ and $\mathbf{X}_{j+1}^{(HH)}$. These two channels will first constitute the inputs of the shared CNN model denoted by C_j. Then, the output of C_j is fed into the two task-specific CNN models illustrated in the upper and lower branches of the network, and designated by $C_j^{(HL)}$ and $C_j^{(LH)}$, respectively. According to Fig. 1, and in addition to the two common input channels used by the shared CNN model, the generation of the vertical detail coefficients $\mathbf{X}_{j+1}^{(LH)}$ relies on a third reference signal corresponding to $\mathbf{X}_{1,j}$. For this

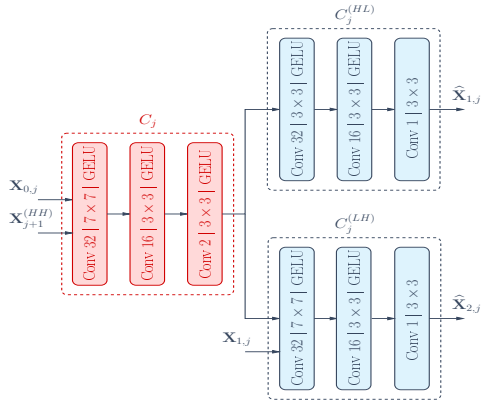


Fig. 4. Proposed MT-CNN model $C_j^{(HL,LH)}$.

reason, an additional channel $\mathbf{X}_{1,j}$ has been included as an input of the CNN model $C_j^{(LH)}$. Finally, the output layers of both task-specific models $C_j^{(HL)}$ and $C_j^{(LH)}$ allows to generate the predicted components $\widehat{\mathbf{X}}_{1,j}$ and $\widehat{\mathbf{X}}_{2,j}$, yielding the horizontal $\mathbf{X}_{j+1}^{(HL)}$ and vertical $\mathbf{X}_{j+1}^{(LH)}$ detail subbands:

$$\begin{cases} \mathbf{X}_{j+1}^{(HL)} = \mathbf{X}_{1,j} - \widehat{\mathbf{X}}_{1,j} \\ \mathbf{X}_{j+1}^{(LH)} = \mathbf{X}_{2,j} - \widehat{\mathbf{X}}_{2,j}. \end{cases} \tag{3}$$

It should be noted here that the structure employed with the shared and task-specific models is inspired by the first CNN-based diagonal prediction stage.

To learn the joint CNN model $C_j^{(HL,LH)}$, a multi-task learning approach will be adopted. Let $\mathbf{\Theta}_j^{(HL,LH)}$ denotes the corresponding vector of parameters given by

$$\mathbf{\Theta}_j^{(HL,LH)} = \left(\mathbf{\Theta}_j, \mathbf{\Theta}_j^{(HL)}, \mathbf{\Theta}_j^{(LH)}\right)^\top \tag{4}$$

where $\mathbf{\Theta}_j$, $\mathbf{\Theta}_j^{(HL)}$ and $\mathbf{\Theta}_j^{(LH)}$ represent the sharing as well as the task-specific parameters. The vector $\mathbf{\Theta}_j^{(HL,LH)}$ is learned by optimizing the sum of the task-specific objective functions. Similarly to the CNN-based diagonal prediction stage, and using a MSE criterion, our multi-task loss function can be expressed as follows

$$\begin{aligned} \mathcal{L}_{1,2}(\mathbf{\Theta}_j^{(HL,LH)}) &= \mathcal{L}_1(\mathbf{\Theta}_j, \mathbf{\Theta}_j^{(HL)}) + \mathcal{L}_2(\mathbf{\Theta}_j, \mathbf{\Theta}_j^{(LH)}) \\ &= \frac{1}{M_j N_j}\Big(\sum_{m=1}^{M_j}\sum_{n=1}^{N_j} (\mathbf{X}_{1,j}(m,n) - \widehat{\mathbf{X}}_{1,j}(m,n))^2 \\ &\quad + \sum_{m=1}^{M_j}\sum_{n=1}^{N_j} (\mathbf{X}_{2,j}(m,n) - \widehat{\mathbf{X}}_{2,j}(m,n))^2\Big). \end{aligned} \tag{5}$$

Once the training is achieved, the learned model can be applied to jointly compute the predicted components $\widehat{\mathbf{X}}_{1,j}$ and $\widehat{\mathbf{X}}_{2,j}$, and deduce the horizontal $\mathbf{X}_{j+1}^{(HL)}$ and vertical $\mathbf{X}_{j+1}^{(LH)}$ detail subbands using (3).

CNN-based update stage

Following the prediction stages, a CNN-based update step is finally performed to compute the approximation coefficients $\mathbf{X}_{j+1}$:

$$\begin{aligned} \mathbf{X}_{j+1} &= \mathbf{X}_{0,j} + \widehat{\mathbf{T}}_j \\ &= \mathbf{X}_{0,j} + C_j^{(LL)}(\widetilde{\mathbf{X}}_{j+1}) \end{aligned} \tag{6}$$

where $\widetilde{\mathbf{X}}_{j+1} = (\mathbf{X}_{j+1}^{(HH)}, \mathbf{X}_{j+1}^{(LH)}, \mathbf{X}_{j+1}^{(HL)})$.

Therefore, the generated detail subbands will constitute the three input channels of the update CNN model $C_j^{(LL)}$, and its output channel $\widehat{\mathbf{T}}_j$ will be used to smooth $\mathbf{X}_{0,j}$ and produce the approximation coefficients $\mathbf{X}_{j+1}$. It should be noted here that the employed $C_j^{(LL)}$ structure is similar to that of $C_j^{(HH)}$ (shown in Fig. 3).

To learn the vector of involved parameters $\mathbf{\Theta}_j^{(LL)}$, we adopt the same optimization technique proposed in [18]. This technique consists in minimizing the error between the ideal low-pass filtered image and the approximation subband $\mathbf{X}_{j+1}$. Thus, the employed loss function is given by

$$\begin{aligned} \mathcal{L}_0(\mathbf{\Theta}_j^{(LL)}) = \frac{1}{M_j N_j}\sum_{m=1}^{M_j}\sum_{n=1}^{N_j} (\mathbf{Y}_{j+1}(m,n) - \mathbf{X}_{0,j}(m,n) \\ - \widehat{\mathbf{T}}_j(m,n))^2 \end{aligned} \tag{7}$$

where $\mathbf{Y}_{j+1}$ is the decimated version of the subband obtained by applying an ideal low-pass filter to the input (i.e approximation) subband $\mathbf{X}_j$.

4. Experimental results

4.1. Experimental settings

The proposed multi-task CNN-based NSLS architecture has been trained using the Flickr dataset composed of 8,000 images with various sizes[1]. While the structure of the involved prediction and update CNN models has been provided in Section 3, the different models have been optimized using the ADAM algorithm [30] with a learning rate of 10^{-3}, a decay of 10^{-4} and a batch size of 8. The training is carried out by using Keras and TensorFlow on an NVIDIA Tesla V100 32 GB GPU. During the testing phase, the different compression schemes were validated using three test datasets. The first one contains 30 samples, of size 1200×1200, taken from the Tecnick sampling dataset[2] [31]. The second one is the popular Kodak dataset, including 24 images of size 768×512[3]. The third one contains 40 crop images, of size 512×512, selected randomly from the Challenge on Learned Image Compression (CLIC) database[4]. Note that the source code of the proposed approach as well as the trained models will be made publicly available.

[1] https://www.kaggle.com/datasets/adityajn105/flickr8k
[2] https://testimages.org/
[3] https://www.r0k.us/graphics/kodak/
[4] http://www.compression.cc/2018/challenge/

4.2. Comparison methods

The proposed approach, which is designated by MT-CNN-LS, is evaluated and compared to different state-of-the-art methods. More precisely, in addition to the JPEG2000 image coding standard, we consider different neural networks-based compression techniques:

- AE-CNN [20] represents an end-to-end optimized image compression method. This method employs an Auto-Encoder (AE) architecture where the encoder is composed of three successive stages of linear (convolution) filter and nonlinear activation functions.
- AE-CNN-Hyp [22] is an extension of the previous method [20], and aims to integrate a hyperprior to exploit the spatial dependencies of the image representation.
- CNN-LS [15] is a recent neural network-based LS. As described in Section 2, the architecture uses CNN for the prediction stage while the update operator is simply replaced by a mean filter.
- FCNN-LS [18] corresponds to our previous work using four different FCNN models to perform the three prediction stages and the update one in a non-separable lifting scheme.

The wavelet-based coding methods (i.e., JPEG2000, CNN-LS [15], FCNN-LS [18] and MT-CNN-LS) are conducted using three resolution levels. Moreover, once the wavelet coefficients are obtained by the different NN-based LS, JPEG2000 has been only used for the entropy encoding.

4.3. Performance metrics

The performance of the aforementioned compression methods is evaluated in terms of Rate-Distortion (R-D). To assess the quality of reconstructed (i.e., decoded) images, different quality metrics could be used. However, according to our previous work [18] as well as the recent quality assessment studies [32, 33], it has been shown that conventional measures (typically PSNR and SSIM [34]) are much less accurate to judge the visual quality improvement of the reconstructed images in the context of neural networks-based image compression methods. This statement will be illustrated later through the subjective results. As a result, many efforts have been recently made to develop new deep learning-based image quality assessment metrics. In our experiments, we propose to use the Perceptual Image-Error Assessment through Pairwise Preference (PieAPP) metric [35], which was found to be better correlated with human perception than its counterparts. In addition, we will illustrate the relative gain of the proposed method in terms of bitrate saving and quality of reconstruction using the Bjøntegaard metric [36].

4.4. Results and discussion

Figures 5(a), 6(a) and 7(a) show the R-D results for the Kodak, Tecnick and CLIC image datasets. Since lower PieAPP values reflect better image quality, it can be first noticed that deep learning-based compression methods improve the JPEG2000 coding standard. Moreover, the recent NN-based lifting schemes (i.e., CNN-LS [15] and FCNN-LS [18]) outperform the remaining state-of-the-art methods. Finally, the proposed MT-CNN-LS-based coding approach leads to the best compression performance. In addition to these average R-D results, Figures 5(b), 6(b) and 7(b) show the R-D plots for three samples taken from the employed test datasets. The obtained plots illustrate the important gain that can be achieved by our MT-CNN-LS method compared to the existing approaches.
Furthermore, the Bjøntegaard metric results of the proposed approach compared to FCNN-LS [18] and CNN-LS [15] are provided in Tables 1 and 2, respectively. The results are obtained at low and middle bitrates given by $\{0.07, 0.1, 0.15, 0.2\}$ and $\{0.2, 0.25, 0.3, 0.4\}$ bits per pixel (bpp), respectively. Note that negative values indicate an improvement in terms of PieAPP as well as bitrate saving. Thus, at the same quality of reconstruction (resp. the same bitrate), the proposed approach leads to a significant gain in terms of bitrate (resp. PieAPP) compared to both reference methods. For instance, for the different test image datasets and at low bitrates, our MT-CNN-LS approach achieves an average bitrate saving of about 15% (resp. 50%) compared to the FCNN-LS [18] (resp. CNN-LS [15]).
A subjective quality assessment of the different NN-based LS and JPEG2000 has also been conducted. For instance, Figures 8 and 9 show some reconstructed images with their associated PieAPP and SSIM metrics. It can be first seen that the proposed MT-CNN-LS approach yields better visual reconstruction quality compared to the other methods. Moreover, while the best SSIM and PieAPP values are highlighted in bold, it can be observed that the PieAPP metric is more appropriate than the conventional SSIM metric and shows more coherent results.
Finally, a complexity analysis in terms of encoding/decoding time and model size (i.e number of parameters) has been carried out for the proposed method as well as the closely related ones CNN-LS [15] and FCNN-LS [18]. Table 3 illustrates this analysis for an image of size 600×600 using an Intel Xeon(R) processor (4 GHz) and a Python implementation. First, it can be noticed that the proposed model as well as the FCNN-LS [18] have similar number of parameters (around 168,000) whereas CNN-LS [15] involves fewer parameters (around 97,000). This difference is mainly due to the fact that CNN-LS [15] uses a single prediction model which is kept fixed at the three resolution levels of the lifting decomposition, whereas the proposed architecture as well as FCNN-LS [18] use prediction and update models specific to each resolution level. Regarding the execution time, it can be observed that our multi-task architecture requires 1.6/0.5 seconds for the encoding/decoding process, which becomes faster than FCNN-LS [18].

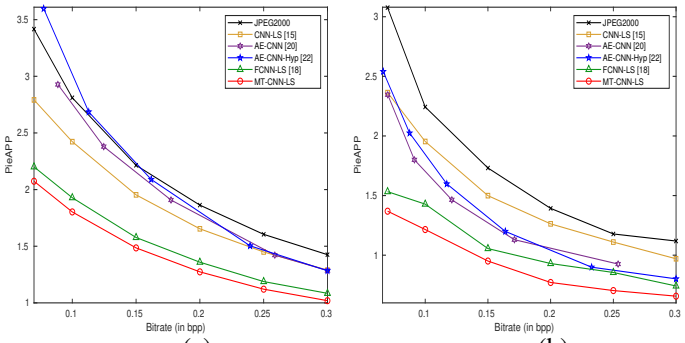
5. Conclusion and perspectives

In this letter, we proposed a novel neural network-based non-separable lifting scheme for image compression purposes. The designed architecture relies on a multi-task CNN model, which aims to perform the horizontal and vertical prediction stages simultaneously. The proposed architecture allows to reduce the number of neural network models employed for the different

lifting stages. However, it does not take into account the dependencies exiting between the diagonal and horizontal as well as vertical prediction steps. The experimental results, obtained with three standard image datasets, have shown the good performance of the proposed approach compared to the state-of-the-art methods, and, more specifically, the recent neural networks-based lifting schemes. In future work, an end-to-end learning strategy could be envisaged to optimize the proposed multi-task CNN architecture.

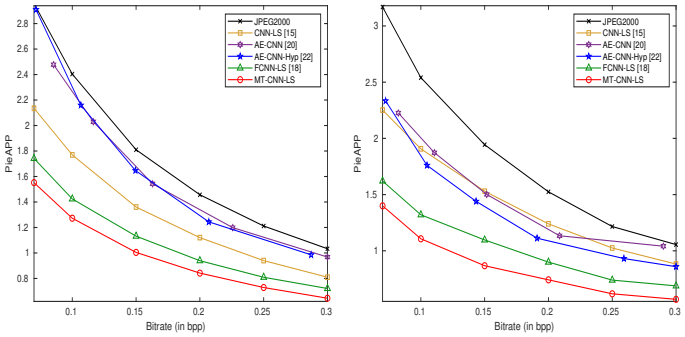
Acknowledgments

This work has received funding from the doctoral school of University Sorbonne Paris Nord, France.



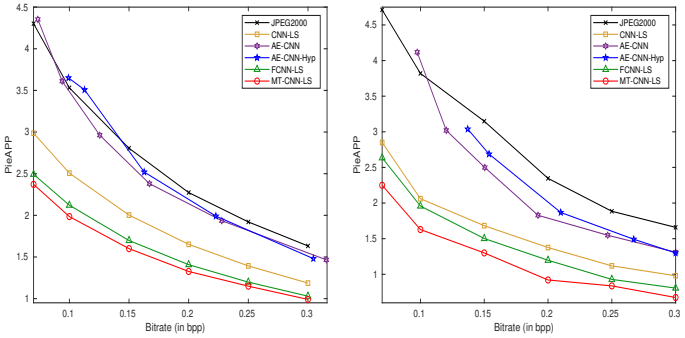
(a) (b)

Fig. 5. R-D results for the Kodak dataset: (a) average results, (b) results for a given image.



(a) (b)

Fig. 6. R-D results for the Tecnick dataset: (a) average results, (b) results for a given image.



(a) (b)

Fig. 7. R-D results for the CLIC dataset: (a) average results, (b) results for a given image.

(a)

(b): PieAPP=3.18, **SSIM=0.74**

(c): PieAPP=2.44, **SSIM=0.74**

(d): PieAPP=1.88, **SSIM=0.74**

(e): **PieAPP=1.73**, SSIM=0.73

Fig. 8. Original test image (a) and reconstructed ones at 0.1 bpp using: (b) JPEG2000, (c) CNN-LS [15], (d) FCNN-LS [18], (e) MT-CNN-LS.

(a)

(b): PieAPP=3.78, SSIM=0.55

(c): PieAPP=2.81, **SSIM=0.58**

(d): PieAPP=2.2, SSIM=0.56

(e): **PieAPP=2.02**, SSIM=0.56

Fig. 9. Original test image (a) and reconstructed ones at 0.15 bpp using: (b) JPEG2000, (c) CNN-LS [15], (d) FCNN-LS [18], (e) MT-CNN-LS.

Table 1. Bjøntegaard metric: the average PieAPP differences and the bitrate saving. The gain of MT-CNN-LS w.r.t FCNN-LS [18]

	bitrate saving (in %)		PieAPP difference	
Datasets	low	middle	low	middle
Kodak	-13.80	-8.74	-0.11	-0.06
Tecnick	-19.87	-14.41	-0.14	-0.07
CLIC	-11.54	-5.13	-0.12	-0.05

.

Table 2. Bjøntegaard metric: the average PieAPP differences and the bitrate saving. The gain of MT-CNN-LS w.r.t CNN-LS [15].

	bitrate saving (in %)		PieAPP difference	
Datasets	low	middle	low	middle
Kodak	-57.01	-26.57	-0.55	-0.23
Tecnick	-51.82	-19.40	-0.44	-0.12
CLIC	-41.77	-22.42	-0.47	-0.22

Table 3. Complexity of the proposed method.

Criterion	CNN-LS[15]	FCNN-LS [18]	MT-CNN-LS
Number of parameters	97,489	167,244	168,546
Encoding time	1.1 s	2.2 s	1.6 s
Decoding time	0.7 s	0.8 s	0.5 s

References

[1] J.-H. Jacobsen, A. W. M. Smeulders, E. Oyallon, *i*-RevNet: Deep invertible networks, in: International Conference on Learning Representations, Vancouver, Canada, 2018, pp. 1–11.

[2] T.-S. Nguyen, M. Luong, M. Kaaniche, L. H. Ngo, A. Beghdadi, A novel multi-branch wavelet neural network for sparse representation based object classification, Pattern Recognition 135 (2023) 109155.

[3] J. J. Huang, P. L. Dragotti, LINN: Lifting inspired invertible neural network for image denoising, in: European Signal and Image Processing Conference, Dublin, Ireland, 2021, pp. 1–5.

[4] Y. Xing, M. Kaaniche, B. Pesquet-Popescu, F. Dufaux, Adaptive non separable vector lifting scheme for digital holographic data compression, Applied Optics 54 (1) (2015) A98–A109.

[5] E. Martinez-Enriquez, J. Cid-Sueiro, F. D. de Mari a, A. Ortega, Directional transforms for video coding based on lifting on graphs, IEEE Transactions on Circuits and Systems for Video Technology 28 (4) (2016) 933–946.

[6] W. Sweldens, The lifting scheme: A custom-design construction of biorthogonal wavelets, Applied and Computational Harmonic Analysis 3 (2) (1996) 186–200.

[7] I. Daubechies, W. Sweldens, Factoring wavelet transforms into lifting steps, Journal of Fourier Analysis and Applications 4 (3) (1998) 247–269.

[8] A. Skodras, C. A. Christopoulos, T. Ebrahimi, JPEG2000: The upcoming still image compression standard, Pattern Recognition Letters 22 (12) (2001) 1337–1345.

[9] A. Gouze, M. Antonini, M. Barlaud, B. Macq, Design of signal-adapted multidimensional lifting schemes for lossy coding, IEEE Transactions on Image Processing 13 (12) (2004) 1589–1603.

[10] M. Kaaniche, B. Pesquet-Popescu, A. Benazza-Benyahia, J.-C. Pesquet, Adaptive lifting scheme with sparse criteria for image coding, EURASIP Journal on Advances in Signal Processing: Special Issue on New Image and Video Representations Based on Sparsity 2012 (1) (2012) 1–22.

[11] J. Solé, P. Salembier, Generalized lifting prediction optimization applied to lossless image compression, IEEE Signal Processing Letters 14 (10) (2007) 695–698.

[12] A. Benazza-Benyahia, J.-C. Pesquet, J. Hattay, H. Masmoudi, Block-based adaptive vector lifting schemes for multichannel image coding, EURASIP International Journal of Image and Video Processing 2007 (1) (2007) 10 pages.

[13] B. Pesquet-Popescu, Two-stage adaptive filter bank, First filling date 1999/07/27, official filling number 99401919.8, European patent number EP1119911, 1999.

[14] M. Kaaniche, A. Benazza-Benyahia, B. Pesquet-Popescu, J.-C. Pesquet, Non separable lifting scheme with adaptive update step for still and stereo image coding, Elsevier Signal Processing: Special issue on Advances in Multirate Filter Bank Structures and Multiscale Representations 91 (12) (2011) 2767–2782.

[15] H. Ma, D. Liu, R. Xiong, F. Wu, iWave: CNN-based wavelet-like transform for image compression, IEEE Transactions on Multimedia 22 (7) (2020) 1667–1697.

[16] H. Ma, D. Liu, N. Yan, H. Li, F. Wu, End-to-end optimized versatile image compression with wavelet-like transform, IEEE Transactions on Pattern Analysis and Machine Intelligence (September 2020).

[17] T. Dardouri, M. Kaaniche, A. Benazza-Benyahia, J.-C. Pesquet, Optimized lifting scheme based on a dynamical fully connected network for image coding, in: IEEE International Conference on Image Processing, Abu Dhabi, United Arab Emirates, 2020, pp. 1–5.

[18] T. Dardouri, M. Kaaniche, A. Benazza-Benyahia, J.-C. Pesquet, Dynamic neural network for lossy-to-lossless image coding, IEEE Transactions on Image Processing 31 (2021) 569–584.

[19] T. Dardouri, M. Kaaniche, A. Benazza-Benyahia, J.-C. Pesquet,

G. Dauphin, A neural network approach for joint optimization of predictors in lifting-based image coders, in: IEEE International Conference on Image Processing, Anchorage, Alaska, USA, 2021, pp. 1–5.

[20] J. Ballé, V. Laparra, E. P. Simoncelli, End-to-end optimized image compression, in: International Conference on Learning Representations, Toulon, France, 2017, pp. 1–27.

[21] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, V. G. Luc, Generative adversarial networks for extreme learned image compression, in: International Conference on Learning Representations, New Orleans, LA, USA, 2019, pp. 1–31.

[22] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, N. Johnston, Variational image compression with a scale hyperprior, in: International Conference on Learning Representations, Vancouver, Canada, 2018, pp. 1–47.

[23] J. Li, B. Li, J. Xu, R. Xiong, W. Gao, Fully connected network-based intra prediction for image coding, IEEE Transactions on Image Processing 27 (7) (2018) 3236–3247.

[24] T. Dumas, A. Roumy, C. Guillemot, Context-adaptive neural network-based prediction for image compression, IEEE Transactions on Image Processing 29 (1) (2019) 679–693.

[25] J. Liu, H. Sun, J. Katto, Learned image compression with mixed transformer-cnn architectures, in: Conference on Computer Vision and Pattern Recognition Workshops, Vancouver, Canada, 2023, pp. 14388–14397.

[26] I. Schiopu, A. Munteanu, Deep-learning based lossless image coding, IEEE Transactions on Circuits and Systems for Video Technology 30 (7) (2019) 1829–1842.

[27] F. Mentzer, L. V. Gool, M. Tschannen, Learning better lossless compression using lossy compression, in: Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 2020, pp. 6637–6646.

[28] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, in: Proceedings of the IEEE, Vol. 86, 1998, pp. 2278–2324.

[29] S. Vandenhende, S. Georgoulis, W. V. Gansbeke, M. Proesmans, D. Dai, L. V. Gool, Multi-task learning for dense prediction tasks: A survey, IEEE Transactions on Pattern Analysis and Machine Intelligence 44 (7) (2022) 3614–3633.

[30] D. P. Kingma, J. L. Ba, Adam: A method for stochastic optimization, in: International Conference on Learning Representations, San Diego, CA, USA, 2015, pp. 1–15.

[31] N. Asuni, A. Giachetti, Test images: A large data archive for display and algorithm testing, Journal of Graphics Tools 17 (4) (2015) 113–125.

[32] G. Valenzise, A. Purica, V. Hulusic, M. Cagnazzo, Quality assessment of deep learning-based image compression, in: International Workshop on Multimedia Signal Processing, Vancouver, BC, Canada, 2018, pp. 1–6.

[33] Z. Cheng, P. Akyazi, H. Sun, J. Katto, T. Ebrahimi, Perceptual quality study on deep learning based image compression, in: IEEE International Conference on Image Processing, Taipei, Taiwan, 2019, pp. 719–723.

[34] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Transactions on Image Processing 13 (4) (2004) 600–612.

[35] E. Prashnani, H. Cai, Y. Mostofi, P. Sen, PieAPP: Perceptual image-error assessment through pairwise preference, in: IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 1–10.

[36] G. Bjøntegaard, Calculation of average PSNR differences between RD curves, Tech. rep., ITU SG16 VCEG-M33, Austin, TX, USA (2001).