PATCH-BASED IMAGE CODING WITH END-TO-END LEARNED CODEC USING OVERLAPPING

Marwa Tarchouli[1,2], Sebastien Pelurson[1], Thomas Guionnet[1], Wassim Hamidouche[2], Meriem Outtas[2], and Olivier Deforges[2]

[1]Ateme, Rennes, France
[2]Univ. Rennes, INSA Rennes, CNRS, IETR - Rennes, France

ABSTRACT

End-to-end learned image and video codecs, based on auto-encoder architecture, adapt naturally to image resolution, thanks to their convolutional aspect. However, while coding high resolution images, these codecs face hardware problems such as memory saturation. This paper proposes a patch-based image coding solution based on an end-to-end learned model, which aims to remedy to the hardware limitation while maintaining the same quality as full resolution image coding. Our methodconsists in coding overlapping patches of the image and reconstruct them into a decoded image using a weighting function.This approach manages to be on par with the performance of full resolution image codingusing anend-to-end learned model, and even slightly outperform it, while being adaptable to different memory size. It is also compatible with any learned codec based on a conv/deconvolutional autoencoderarchitecture without having to retrain the model.

KEYWORDS

Auto-encoders, Image compression, Deblocking, block artifacts.

1. INTRODUCTION

During the past few years, deep learning based end-to-end image and video coding field achieved great improvements, and performance of such algorithms can now compete with traditional coding systems like JPEG[1], JPEG2000[2] or BPG [3].

Their auto-encoder architecture, built with convolutional layers, enables processing different image resolutions, no matter the resolution used during the training step. However, with growing model sizes or picture and video resolutions (4K, 8K), these solutions face hardware memory saturation. For example, coding a standard resolution such as an HD image using a powerful GPU as NVIDIA GeForceRTX 2080ti with a memory capacity of 11Go, is not possible as it does not fit into the memory requirement. For4K resolution, it is of course worse.

One way to solve this issue is to use a patch-based coding approach. The image is divided into patches having the same size that can be encoded independently. The patch size should be smaller than the image size and should fit into the memory constraints. Then, the decoded patches are gathered to reconstruct the decoded image, as illustrated in Fig.*1*. Moreover, this method enables to implement several forms of parallelization as well as the ability to access a sub-part of the image without entirely decoding it.

This solution addresses the hardware limitation issues, but the reconstructed picture can have block artifacts in the patch boundaries, widely deteriorating the image quality (Fig.*2*).

David C. Wyld et al. (Eds): AIAA, DPPR, DSA, ICSS, IOTE, NLPTA, WEST, CACIT, BMLI- 2022
pp. 53-63, 2022. CS & IT - CSCP 2022 DOI: 10.5121/csit.2022.122305

In this paper, the main goal is to address the hardware limitation issue of end-to-end learned codecs without deteriorating the decoded image quality (i.e. remove block artifacts). To do so, overlapping patches are encoded and then a weighting function is used to reconstruct the overlapped pixels. This method provides an objective and subjective quality slightly higher than the full image encoding approach, while leveraging the memory consumption flexibility of the patch-based coding solutions. However, it comes with a slight increase in complexity.

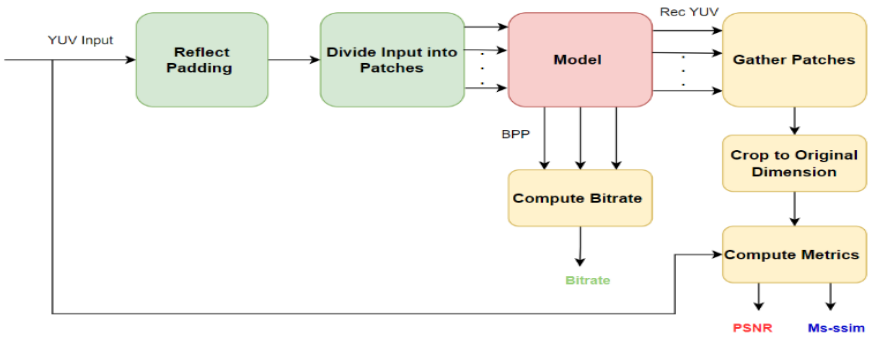


Fig.1 Steps to patch-based image coding

The rest of the paper is structured as follows. In section 2, the related work on end-to-end learned image coding as well as approaches to eliminate block artifacts are presented. Then, the patch-based image coding using overlapping method is explained in section 3. Section 4 presents the results of the driven experiments. Finally, section 5 concludes the paper.

Fig.2 Reconstructed image per patch using an end-to-end learned codec

2. RELATED WORK

2.1. End-To-End Learned Image Compression

Recently, deep learning codecs have accomplished promising results in a short period of time. In fact, end-to-end image compression managed to achieve the state-of-the-art of traditional image coding in terms of coding efficiency. Authors in[4] proposed an image compression structure based on a conv/deconvolutional auto-encoder [5] whose role is to transform the input image into a latent representation. Then, a factorized entropy model is used to estimate the probability distribution of this representation allowing entropy coding. Balléet al.[6] replaced the factorized entropy model by a hyperprior auto-encoder. This mechanism enables the probability model to

adapt itself to the input image as well as to capture spatial dependencies in the latent representation.

This work is considered as the reference design for numerous other state-of-the-art contributions. In the same context,[7] uses an autoregressive module to improve the entropy model. It succeeds in improving the coding performance compared to[6]. However, this improvement leads to an increase in complexity.[8] presented a model architecture based on residual blocks [9] and attention modules in order to extract a compact and an efficient latent representation. In addition, it presented an entropy model which exploits Gaussian mixture model to parameterize the latent distributions. This latter method achieves performances that compete with the versatile video coding (VVC) [10]encoder in Intra mode.

Some works [11] used the ability of generative adversarial network (GAN) [12]to generate realistic and sharp images for image compression at low bitrate and for small resolutions, while [13], [14]combined adversarial loss with rate-distortion loss to include the fidelity aspect into the network allowing better subjective quality for high resolution images.

Recurrent neural networks (RNN) are used in some approaches[15], [16]. They present progressive models with architectures designed on top of residuals. These approaches achieve results better than JPEG. However, they perform worse than conv/deconvolutional auto-encoder solutions with entropy models like [8], while having a higher complexity.

The auto-encoder solution presented in [8] achieves the best performance among learned image codecs. It is competitive with VVC, the latest traditional codec. Despite the performance of this solutions, they require special hardware to operate effectively (GPU), which is limited in memory capacity.

2.2. Deblocking with Neural Networks

Deep learning approaches were also explored for post-processing decoded images, particularly to eliminate block artifacts. Some works focused on enhancing quality of images coded by traditional codecs. For instance, [17] provides a network architecture to remove block artifacts from JPEG compressed blocks using neighbouring blocks, while[18] tempted to deblock the whole decoded image. [19]introduced pre & post-processing networks to improve JPEG compression performance. In fact, [19]proposed an end-to-end framework composed of two convolutional neural networks (CNN) along with a handcrafted image codec such as JPEG, JPEG2000 or BPG. The first CNN learns an optimal and a compact representation of the image, which is compressed by the traditional image codec. Then, the second CNN enhances the quality of the decoded image. The results of[19] outperform several state-of-the-art methods of image deblocking and denoising. However, this work was only tested on grayscale, images with small resolutions, which is not compatible with real compression applications.

Other works were interested in deblocking images coded by learned image codecs, especially progressive learned codecs. For instance,[20] proposed a patch-based image coding framework called BINet which uses binarize dneigh bouring patches to eliminate block artifacts. Each patch is reconstructed using 9 surrounding binarizedneighbours. This approach outperforms JPEG at low bitrates.

[21] and [22] address the same problem by introducing a postprocessing network to remove blocking artifacts, which increases the size of the model and the complexity of the training process.

With a different objective, super resolution models, such as VDSR [23], showed promising results in enhancing decoded image quality.

Nevertheless, all these related works are either networks which require a dependence on traditional image codecs, or their training process is difficult, and retraining is necessary if the learned codec changes, or, they are not consistent with codecs based on conv/deconvolutional auto-encoders architectures such as[8]. This paper proposes a patch-based image coding approach to address the hardware limitation of any conv/deconvolutional learned image codec without requiring a retraining. It achieves the same quality level as the full resolution coding and enables a flexible use of memory which make the method undemanding in terms of hardware material. However, these results are achieved at the expense of a slight increase of complexity (3%).

3. PATCH-BASED IMAGE CODING USING OVERLAPPING

3.1. Proposed Method

In this approach, the process described in
Fig.*1* is followed. First, reflect padding is applied to the input image in order to make its size divisible by the patch size. While dividing the image into patches, every two consecutive patches must have a range of pixels in common horizontally and vertically, as it is illustrated in
Fig.*3*. Then, the coding step is performed by an end-to-end learned image codec on input patches of size $P + N$, where P is the size of the patch w/o overlapping and N is the number of the overlapped pixels. In the next step, the image is reconstructed from the patches. The overlapping areas are combined by a weighting function which generates a progressive transition from one patch to the other. In fact, if b_m andb_{m+1} are two consecutive reconstructed patches overlapping horizontally on$N$ pixels, the value of the $i^{t\square}$ overlapped pixel $p_{rec}(i)$, for a given line in the reconstructed image is determined by the following equation:

$$p_{rec}(i) = \left(1 - \frac{i}{N-1}\right) p_{b_m}(P+i) + \left(\frac{i}{N-1}\right) p_{b_{m+1}}(i), \quad (1)$$

where $i \in \{0, \ldots, N-1\}$ is the index of the overlapped pixels, P is the size of the patch w/o overlapping, p_{b_m} and p_{bm+1} are pixels values, for a specific line, of two consecutive decoded patches b_m and b_{m+1} respectively.

The same equation is valid for vertically overlapping patches. Once the decoded image is reconstructed, quality metrics can be computed.

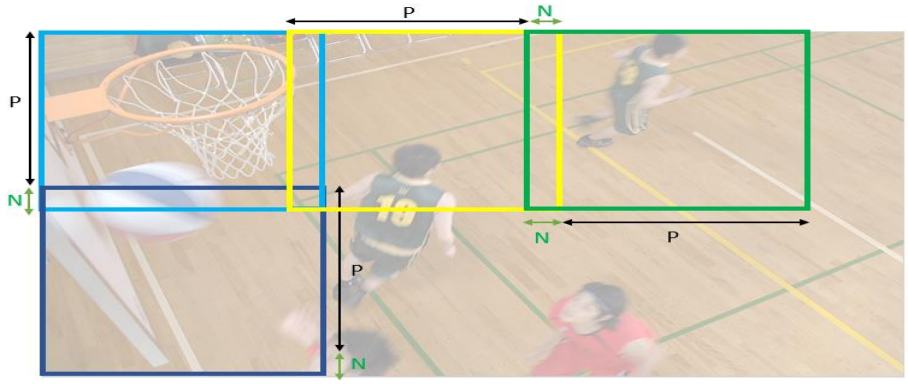


Fig.3 Method to overlap patches on N pixels

3.2. Training And Evaluation

The end-to-end learned codec used in this paper is an implementation of the model architecture introduced in [8]. It is a conv/deconvolutional auto-encoder model leveraging the efficiency of residual blocks and attention modules, in addition to using a Gaussian Mixture model as the entropy engine. It was trained on 400 000 samples from the CLIC 2020 dataset [24]. For training, 256x256 sized patches were randomly cropped from each image of the training set. The model was trained on a total of 500 000 steps. It was optimized using an Adam optimizer, with a batch size of 4. The learning rate value was set to 10^{-4} for the first 200 000 steps and then it was decreased to $10^{-5}$for the rest of the training. The loss function to be minimized is the rate-distortion loss function formulated as :

$$J = D + \lambda R, \quad (2)$$

where Drefers to the distortion between the original patch and the reconstructed one, measured by the Mean Square Error (MSE) or the Multi-Scale Structural Similarity Index**Error! Reference source not found.** (MS-SSIM) metrics, and R refers to the rate used to transmit the bitstream, estimated using the Shannon entropy. λ is the Lagrangian multiplier, allowing to adapt the bit rate targeted by the learned image coding model. The goal is the teach the model to minimize the distortion between the original patch and the reconstructed one while using a reasonable bit rate.

Eight models have been trained, 4 for each quality metric (MSE and MS-SSIM), matching 4 different bit rates. The corresponding Lagrangian multipliers are λ ={420, 220, 120, 64} for MS-SSIM models and λ = {4096, 3140, 2048, 1024} for MSE models.

Our method is then evaluated on Class B, C, D, E and F of the JVET Common Test Conditions (CTC) sequences (8 bit sequences), which have different resolutions (HD, 1280x720, 840x832 and 416x240). For each sequence, one frame is extracted and compressed both entirely (referred as the full image approach) and by the proposed patch-based approach, with and without overlapping. We used N € {0, 2, 4, 8, 16, 32} overlapped pixels and we set $P = 256$ similar to the training resolution.

4. EXPERIMENT RESULTS

4.1. Complexity And Memory Analysis

Our method allows using the available GPU memory in a flexible way by coding multiple patches simultaneously. In fact, instead of coding the largest possible patches sequentially, we explored the GPU ability to parallelize processing. Therefore, the model was fed a batch of patches as input. The input shape becomes then [B, C, H, W] where C, H and W refer, respectively, to the channel number, the height, and the width of the input patch while B corresponds to the number of patches processed in parallel, called the batch size.

While fixing the patch size (W and H), the batch size value can be adapted to the available GPU memory. Hence, if the total number of patches to code cannot be fed to the model as one batch, the input patches are regrouped to smaller batches.

Table 1 compares the coding time of our proposed approach using parallelization, with the coding time of the full resolution learned coding. For information purposes, the coding time of two more methods is computed: our approach without parallelization and the patch-based learned image coding without overlapping using parallelization.

To achieve this experiment, images of different resolutions were extracted from the JVET CTC and were coded using two machines with two powerful GPUs: Nvidia GeForce RTX 2080ti and Nvidia GeForce RTX 3090 with memory capacity of 11Go and 24Go respectively.

Full resolution coding of an HD image is not possible on both GPUs due to Out Of Memory (OOM) error, while full coding an 1280x720 image, can only be run on the machine with the GPU RTX 3090 since it has more available memory (24Go). It is important to note that these resolutions are standard resolutions in practical applications of image compression. Therefore, the fact that they cannot be run on one of the latest GPUs is inconvenient. In this case, our method provides a solution that enables coding high resolution images without deteriorating quality.

While our method is necessary to code high resolution images (HD, 720p, ...), it is adding more complexity to the system for smaller resolutions. For instance, when running the resolution 832x480 on 2080ti GPU, patch-based coding increases the coding time by 3.63% compared with full resolution coding. This is expected since our method requires coding more pixels in order to overlap patches. Patch-based coding w/o overlapping confirms this explanation since it has approximately the same coding time as full resolution coding.

Table 1 : Coding Time

Resolution	Method	Coding time GPU 2080 11Go	Coding time GPU 3080 24Go	Total Number of patches	Batch size	N
HD	Full resolutioncoding	OOM*	OOM*	-	-	-
	Patch coding in parallel w/o overlapping	3.40s	1.96s	40	8	-
	Patch coding in parallel with overlapping	3.82s	2.05s	40	8	16
	Patchcoding sequentially with overlapping	6.15s	2.86s	40	-	16
1280x720	Full resolutioncoding	OOM*	0.93s	-	-	-
	Patch coding in parallel w/o overlapping	1.75s	0.95s	15	5	-
	Patch coding in parallel with overlapping	1.91s	1.01s	15	5	16
	Patch coding sequentially with overlapping	2.73s	1.25s	15	-	16
832x480	Full resolutioncoding	1.06s	0.52	-	-	-
	Patch coding in parallel w/o overlapping	1.05s	0.54s	8	8	-
	Patch coding in parallel with overlapping	1.10s	0.55s	8	8	16
	Patch coding sequentially with overlapping	1.586s	0.70s	8	-	16

* "OOM" stands for “Out Of Memory”

To conclude this section, our approach addresses the hardware limitation problem since it allows coding resolutions such as HD and 720p. Moreover, it enables flexible adaptation to available memory. However, it increases slightly the complexity of the coding system. We believe this increase in complexity is worthwhile for the solution this method proposes, especially for coding high resolution images.

4.2. Rate-Distortion and Visual Results

Table 2 sums up the BD-rate gain of the patch-based learned image coding with and without overlapping comparing to learned full image coding on CTC sequences, using an end-to-end model trained to minimize MS-SSIM as distortion metric. For the purpose of this evaluation,full image approach for resolutions such as HD and 720p,was run on CPU. It is important to note that using CPU to run an end-to-end leaned codec is not considered because itis severely time-consuming.

Patch-based image coding without overlapping and with 2 overlapped pixels (N = 2) presents a loss in BD-rate, comparing to full image coding. The first observation confirms the block artifacts issue caused by patch-based approaches. The second one is explained by the fact that two overlapped pixels are not enough to recover from these artifacts. Yet, it did eliminate most of them since the BD-rate loss of our proposed method with N=2 is decreased significantly compared to patch-based coding without overlapping. As the number of overlapped pixels increases, the loss in BD-rate decreases which indicates that the block artifacts are removed efficiently. Hence, our method has managed to be on par with the performance of full image coding.

In fact, with N = 8 and N = 16, marginal gains are observed compared to full image coding. As the end-to-end coding model was trained on 256x256 cropped images, it performs better on coding patches of similar size than coding the full resolution image, which explains the gain in BD-rate. For N > 16, BD-rate gains saturation is observed.

Table 2 : BD-rate (MS-SSIM) gains (%) of patch-based coding schemes compared to full image coding system for CTC sequences.

	Sequence	Patch w/o Overlapping	Patch with Overlapping				
			N = 2	N = 4	N = 8	N = 16	N = 32
Class B	Cactus	0.573	-0.0003	-0.033	-0.066	-0.083	-0.079
	BasketballDrive	0.630	0.062	0.024	-0.013	-0.032	-0.030
	BQTerrace	0.744	0.025	-0.020	-0.060	-0.083	-0.090
Class C	RaceHorses	0.503	0.013	-0.014	-0.041	-0.052	-0.048
	BasketballDrill	0.484	0.022	-0.004	-0.034	-0.049	-0.051
	BQMall	0.732	0.056	0.014	-0.023	-0.044	-0.038
	PartyScene	0.428	0.032	0.005	-0.020	-0.032	-0.027
Class D	BasketballPass	0.286	0.025	0.010	-0.012	-0.015	-0.009
	BlowingBubbles	0.146	0.019	0.006	-0.008	-0.016	-0.012
	BQSquare	0.214	0.020	0.008	-0.005	-0.011	-0.012
	RaceHorses	0.281	0.026	0.009	-0.011	-0.022	-0.022
Class E	Johnny	1.240	0.081	0.041	-0.004	-0.026	-0.029
	FourPeople	0.643	0.030	-0.012	-0.046	-0.068	-0.072
	KristenAndSara	0.991	0.062	0.021	-0.015	-0.043	-0.043
Class F	BasketballDrillTex t	0.506	0.035	0.010	-0.019	-0.035	-0.035
	SlideShow	1.018	0.081	0.028	-0.006	-0.031	-0.031
	SlideEditing	0.533	0.019	-0.006	-0.030	-0.035	-0.031

Rate-distortion curves for MS-SSIM models are reported in Fig.5(a) and confirm previous observations. The patch-based coding approach with overlapping allows to fill the gap with the full image coding one. While reaching a better quality, the proposed method increases slightly the rate comparing to patch-based coding without overlapping, which is expected as more pixels are

encoded. We can also conclude that the overlapping method with 8 overlapped pixels provides the best results. In fact, the decoded images of the methods mentioned before are visualized in Fig.4.Overlapping with N = 2 (Fig.4.d) and N = 4 (Fig.4.e) reduces the block artifacts while overlapping with 8 pixels (Fig.4.f) eliminates them entirely.

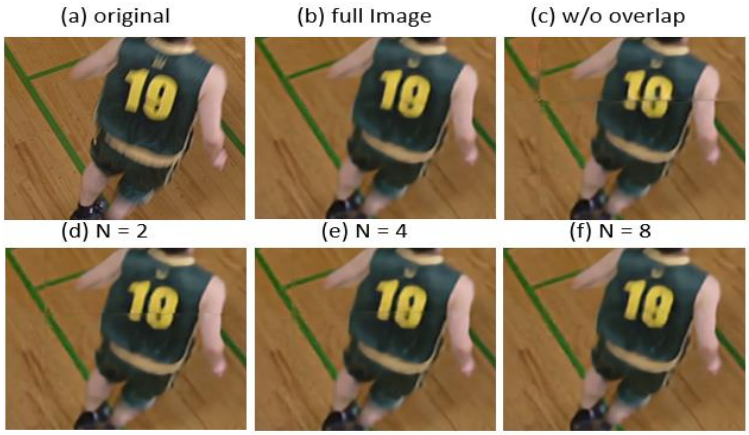
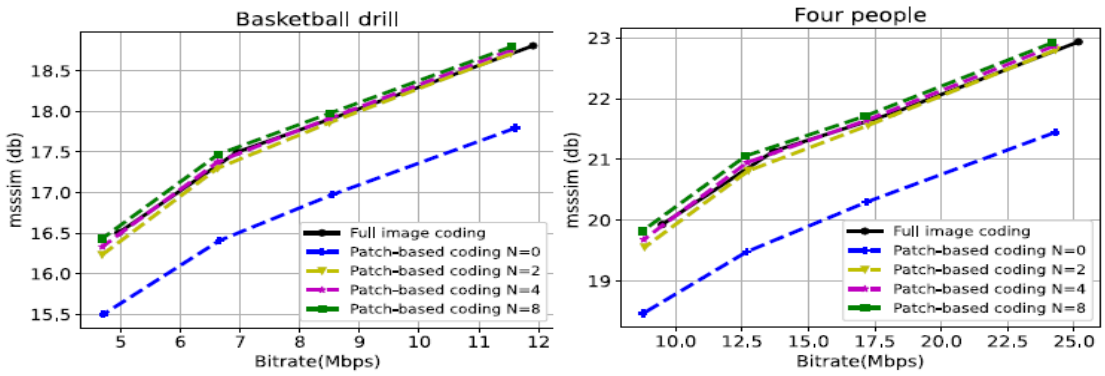
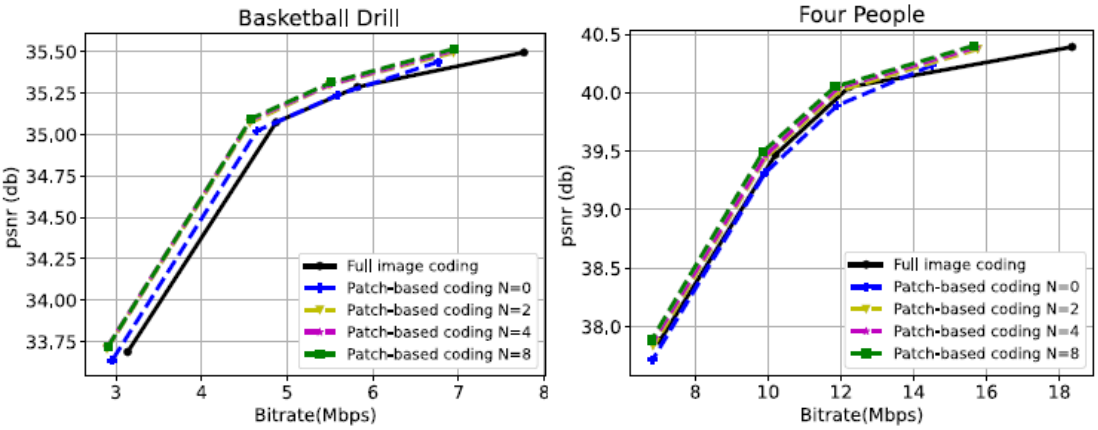


Fig.4 Visual results for BasketballDrill using MS-SSIM model with $\lambda = 420$



(a) MS-SSIM Rate-Distortion results



(b) PSNR Rate-Distortion results

Fig.5 Rate-Distortion results of the proposed approach compared to Full Image coding and patch-based image coding without overlapping for Basket ball Drill and Four People sequences.

The BD-rate results with distortion measured with Peak Signal to Noise Ratio (PSNR) are provided in Table 3,for the end-to-end learned codec trained to optimize the MSE metric. Compared with full image coding, patch-based image coding without overlapping showed a BD-rate loss for PSNR metric (Table 3) less important than the BD-rate loss for MS-SSIM metric (Table 2).This is also illustrated in Fig.5(b) as the quality gap between image coding per patch without overlapping and the full image coding has narrowed. Furthermore, although block artifacts existin the decoded image (Fig.6.c), they are less visible than the decoded image resulting from the MS-SSIM end-to-end codec (Fig.4.c). This behavior may be because MS-SSIM is computed using a sliding window while MSE is a pixel-based metric. In any case, the proposed approach improves the BD-rate performance, the rate-distortion curves (Fig.5) and the perceptual quality of the decoded image (Fig.6 and Fig.4).

Table 3 : BD-rate (PSNR) gains (%) of patch-based coding schemes compared to full image coding system for CTC sequences.

	Sequence	Patch w/o Overlapping	Patch with Overlapping				
			N = 2	N = 4	N = 8	N = 16	N = 32
Class B	Cactus	-0.022	-0.073	-0.080	-0.086	-0.096	-0.089
	BasketballDrive	0.042	-0.010	-0.010	-0.010	-0.013	-0.010
	BQTerrace	-0.040	-0.100	-0.113	-0.121	-0.132	-0.123
Class C	RaceHorses	0.010	-0.033	-0.037	-0.045	-0.046	-0.040
	BasketballDrill	-0.030	-0.071	-0.077	-0.078	-0.079	-0.070
	BQMall	0.013	-0.018	-0.023	-0.020	-0.021	-0.010
	PartyScene	0.033	-0.001	-0.006	-0.010	-0.016	-0.020
Class D	BasketballPass	-0.014	-0.047	-0.046	-0.050	-0.040	-0.022
	BlowingBubbles	0.005	0.001	0.002	-0.008	-0.014	-0.012
	BQSquare	0.017	0.008	0.007	0.014	0.015	0.020
	RaceHorses	0.009	-0.005	-0.010	-0.013	-0.017	-0.017
Class E	Johnny	0.049	0.016	-0.002	-0.013	-0.023	-0.010
	FourPeople	0.016	-0.023	-0.031	-0.045	-0.070	-0.070
	KristenAndSara	0.042	-0.001	-0.011	-0.026	-0.058	-0.030
Class F	BasketballDrillText	0.011	-0.031	-0.036	-0.040	-0.042	-0.040
	SlideShow	0.051	-0.030	-0.032	-0.026	-0.034	-0.020
	SlideEditing	0.029	-0.006	-0.009	-0.016	-0.016	-0.010

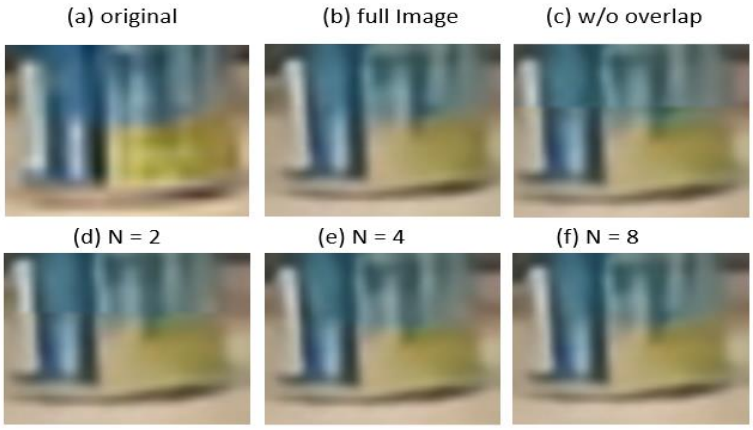


Fig.6. Visual results for Four People using MSE model with $\lambda = 4096$

5. CONCLUSION

This work proposes a solution to the memory saturation problem that end-to-end learned codecs face while coding high resolution images such as HD. This solution consists in patch-based image coding while removing block artifacts using overlapping. This method benefits from flexible memory consumption and manage not only to achieve full image coding performance, but also to improve it slightly (-0.034% for MSE models and -0.024% for MS-SSIM models), even though it increases lightly the complexity (3%). Results are provided on JVET CTC sequences with MSE and MS-SSIM based models. The network architecture of[8] is used as our baseline end-to-end learned codec, but this method is compatible with any learned codec based on a conv/deconvolutional auto-encoder architecture. Furthermore, the proposed method also applies to other image processing tasks, such as denoising, thus enabling patch-based processing for a wide range of CNN based applications.

In our future works, we aim to remedy to the increase in complexity, caused by coding additional overlapped pixels, by training an end-to-end image coding model to smooth the patch borders and hence remove the border artifacts automatically.

REFERENCES

[1] G. K. Wallace (1991) “The JPEG Still Picture Compression Standard,”,IEEE Transaction on Consumer Electronict, pp. 1—17

[2] C. Christopoulos & A. Skodras &T. Ebrahimi (2000)“The jpeg2000 still image coding system: an overview”, vol. 46, pp. 1103-1127.

[3] F. Bellard, “BPG Image format https://bellard.org/bpg/”.

[4] J.Ballé & V. Laparra & E. P Simoncelli(2017)“End-to-end optimized image compression” ICLR 2017 - Conference Track Proceeding.

[5] "P. Vincent & H. Larochelle & Y. Bengio& P.-A. Manzagol (2008) “Extracting and composing robust features with denoising autoencoders”, Intl. conf. on Machine Learning (ICML) .

[6] J. Ballé& D. Minnen& S. Singh &S.J Hwan, N.Johnston (2018)“Variational image compression with a scale hyperprior” ICLR 2018 - Conference Track Proceedings.

[7] J. Ballé & D. Minnen & G.Toderici (2018) “Joint Autoregressive and Hierarchical Priors for Learned Image Compression” Advances in Neural Information Processing Systems, pp. 10771-10780.

[8] Z. Cheng & H. Sun & M. Takeuchi & J. Katto (2020) "Learned image compression with discretized gaussian mixture likelihoods and attention modules" IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7936-7945.

[9] K. He & X. Zhang & S. Ren & J. Sun (2016)“Deep Residual Learning for Image Recognition” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770-778.

[10] G. J. Sullivan& J. R. Ohm (2018)“Versatile video coding Towards the next generation of video compression” Picture Coding Symposium, vol. Jun-2018.

[11] S. Santurkar & D. Budden & N. Shavit (2018)“Generative Compression” Picture Coding Symposium (PCS), pp. 258-262.

[12] I. Goodfellow & J. Pouget-Abadie & M. Mirza & B. Xu & D. Warde-Farley & S. Ozair & A. Courville & Y. Bengio(2014)“Generative adversarial nets” Advances in Neural Information Processing Systems, vol. 3, pp. 2672-2680.

[13] F. Mentzer & G. Toderici& M. Tschannen& E. Agustsson(2020)“High-fidelity generative image compression” Advances in Neural Information Processing Systems, vol. 33, pp. 11913--11924.

[14] E. Agustsson & M. Tschannen & F. Mentzer & R. Timofte & L. Van Gool (2019) “Generative Adversarial Networks for Extreme Learned Image Compression” IEEE/CVF International Conference on Computer Vision (ICCV), pp. 221-231.

[15] G. Toderici& D. Vincent & N. Johnston & S. Hwang & D. Minnen& J. Shor & M. Covell (2017) “Full resolution image compression with recurrent neural networks” CVPR 2017, vol. 2017, pp. 5435-5443.

[16] G. Toderici & S. M. O'Malley & S. Hwang& D. Vincent & D. Minnen & S.Baluja& M. Covell & R. Sukthankar (2016) "Variable rate image compression with recurrent neural networks" ICLR - Conference Track Proceedings.

[17] D. Maleki & S. Nadalian & M. M. Derakhshani & M. A. Sadeghi (2018) "BlockCNN: A Deep Network for Artifact Removal and Image Compression" Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops.

[18] L. Ke & B. Bahetiyaer & Y. Bo (2017)"An Efficient Deep Convolutional Neural Networks Model for Compressed Image Deblocking" Conference, IEEE International, pp. 1320--1325.

[19] F. Jiang & W. Tao & S. Liu & J. Ren & X. Guo & D. Zhao (2018)"An End-to-End Compression Framework Based on Convolutional Neural Networks" IEEE Transactions on Circuits and Systems for Video Technology, vol. 28, pp. 3007--3018.

[20] A. Nortje & W. Brink & H. Engelbrecht & H. Kamper (2021)"BINet: A binary inpainting network for deep patch-based image compression" Signal Processing: Image Communication, vol. 92.

[21] W. Yaojun& L. Xin. & Z. Zhizheng& J. Xin & C. Zhibo (2022) "Learned Block-Based Hybrid Image Compression" IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, pp. 3978-3990.

[22] Z. Zhenghui. & J. Chuanmin. & W. Shanshe& M. Siwei&Y. Jiansheng (2021)"Learned Image Compression Using Adaptive Block-Wise Encoding and Reconstruction Network" IEEE International Symposium on Circuits and Systems (ISCAS), pp. 1-5.

[23] J. Kim. & J. Lee. & K. Lee (2016)"Accurate image super-resolution using very deep convolutional networks" CVPR, pp. 1646--1654.

[24] "Workshop and C. on Learned Image Compression"Https://www.compression.cc/.

[25] Z. Wang& E. P. Simoncelli& A. C. Bovi (2003), "Multiscale structural similarity for image quality assessment," The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, Vol.2, pp. 1398-1402