Advances in video compression: a glimpse of the long-awaited disruption

Thomas Guionnet
Marwa Tarchouli
Sébastien Pelurson
Mickaël Raulet

CITE THIS ARTICLE

Guionnet, Thomas; Tarchouli, Marwa; Pelurson,Sébastien; Raulet, Mickaël ; 2022. Advances in video compression: a glimpse of the long-awaited disruption. SET INTERNATIONAL JOURNAL OF BROADCAST ENGINEERING. ISSN Print: 2446-9246 ISSN Online: 2446-9432. doi: 10.18580/setijbe.2022.3. Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

Advances in video compression: a glimpse of the long-awaited disruption

Thomas Guionnet, Marwa Tarchouli, Sébastien Pelurson and Mickaël Raulet

Ateme, {t.guionnet, m.tarchouli, s.pelurson, m.raulet}@ateme.com

Abstract— **The consumption of video content on the internet is increasing at a constant pace, along with an increase of video quality. As an answer to the ever-growing demand for high quality video, compression technology improves steadily. About every decade, a new major video compression standard is issued, providing a decrease of bitrate by a factor two. Interestingly, the technology does not change radically between codecs generations. Instead, the same block-based hybrid video coding scheme principles and ideas are re-used and pushed further. All along the video compression history, there were several attempts to depart from this model, but none achieved to be competitive. Following the latest codec generation, VVC, the research community has started focusing on deep learning-based strategies. Could it be the new contender to the classical hybrid approach? This paper analyzes the benefits and limitations of deep learning-based video compression methods, and investigates practical aspects such as rate control, delay, memory consumption and power consumption. Overlapping patch-based end-to-end video compression strategy is proposed to overcome memory consumption limitations.**

Index Terms—**Video Compression, Video codec, MPEG-2, H.264, AVC, HEVC, VVC, artificial intelligence, machine learning, deep learning, end-to-end video encoding.**

I. Introduction

THE consumption of video content on the internet is increasing at a constant pace, along with an increase of video quality. Cisco [1] estimates that by 2023, two-thirds of the installed flat-panel TV sets will be UHD, up from 33 percent in 2018. The bitrate for 4K video is more than double the HD video bitrate, and about nine times more than SD bitrate. As an answer to the ever-growing demand for high quality video, compression technology improves steadily. Video compression is a highly competitive and successful field of research and industrial applications. Billions of people are impacted, from TV viewers and streaming addicts to professionals, from gamers to families. Video compression is used for contribution, broadcasting, streaming, cinema, gaming, video-surveillance, social networks, videoconferencing, military, you name it.

The video compression field stems from the early 80's. Since then, it has grown continuous improvements, and strong attention from the business side - the video encoder market size is planned to exceed USD 2.2 Billion by 2025 [2]. About every decade, a new major video compression standard allows halving the required bitrate to achieve a given quality. The latest milestone is the Versatile Video Coding (VVC) standard, issued in 2020. From generation to generation, until VVC, coding efficiency has been improved by relying on the same principle, that is, the block-based hybrid video coding scheme [4]. For more than 30 years, the video compression field has known no revolution or disruption. Instead, the same principles and ideas have been re-used and pushed further. At each generation, existing tools are enhanced, new local coding tools are added, but the overall structure remains the same. In other words, each generation is a complexified version of the previous one. The algorithmic complexity increase is directly reflected by the implementation complexity. For instance, the VVC verification software model (VTM) is about 10 times slower than its predecessor, the High Efficiency Video Coding (HEVC) verification model (HM). Many attempts have been made to depart from the block-based hybrid scheme, none of them have been successful so far.

As of today, the tremendous progression of video compression technology is not compensating for the increase in the demand for always more and higher quality video services. Therefore, the research effort is still ongoing, seeking improvements over VVC, as it was over each previous codec generation. Indeed, The Joint Video Expert Team (JVET), a working group managed by both ISO/IEC MPEG and ITU-T VCEG international standardization bodies, responsible for the development and support of VVC, is currently conducting explorations beyond VVC. There is a new situation arising though: this exploration is following two distinct tracks. One is "classical", consisting in adding or enhancing coding tools to VVC, while the other is dedicated to the exploration of the usage of machine learning (ML). The field of ML, and more particularly deep learning (DL), has made dramatic advances during the last decade, especially in the computer vision domain. There are several ways of applying ML to video compression. One can consider creating elementary coding tools, replacing, or complementing the existing tools in the hybrid block-based scheme. At the other extremity of the spectrum, one can completely replace the hybrid block-based scheme by a deep learning model. The latter solution is highly disruptive with respect to the current video compression history. Hence the question: to what extent is ML becoming essential to video compression?

The goal of this paper is to analyze the benefits and limitations of deep learning-based video compression methods, and to investigate practical aspects such as rate control, delay, memory consumption and power

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

consumption. In a first part, the evolution of video compression is recounted, with a few words on previous attempts to depart from the hybrid block-based model. In a second part, the deep-learning strategies are described, with a focus on tool-based, end-to-end, and super-resolution-based strategies. In a third part, the practical limitations for industrial applications are analyzed. Finally, a technology is proposed, namely overlapping patch-based end-to-end video compression, to overcome memory consumption limitations. Experimental results are provided and discussed.

II. A SHORT HISTORY OF VIDEO COMPRESSION

A. *CODECs and applications*

The idea of temporal prediction for video compression can be tracked back to 1929, with a patent advocating the coding of successive image differences [3], but the modern history of video compression really starts in the 80's. Two organizations are essentially responsible for video coder/decoder (codec) standardization [5][6]: the International Telecommunications Union – Telecommunication Standardization Sector (ITU-T) Video Coding Expert Group (VCEG), a United Nations Organization (formerly CCITT) [7], and the International Organization for Standardization and International Electrotechnical Commission (ISO/IEC) Moving Picture Expert Group (MPEG). ISO is an independent, non-governmental international organization with a membership of 167 national standards bodies [8]. Aside from standardization, many proprietary or independent codecs exists. Nonetheless, the most successful and well-known line of codecs stems from standardization and constitutes the focus of this paper.

The first standardized video codec, ITU-T H.120 [63], has been issued in 1984, then updated in 1988. It already includes a form of intra prediction (Digital Pulse Coded Modulation, DPCM), scalar quantization, entropy coding in the form of variable length coding (VLC) and motion compensation.

ITU-T H.261 [64] was first issued in 1988. It is dedicated to video telephony and introduces the most important block-based motion compensation and Discrete Cosine Transform (DCT). It is the first practically successful video codec. It was later replaced by the dramatically improved ITU-T H.263 [65] in 1995.

Meanwhile ISO/IEC developed MPEG-1 [66], issued in 1993. It was designed to compress VHS-quality raw video, thus enabling first digital TV applications (videos CD, Cable, satellite). One may note that the best-known part of MPEG-1 is the MP3 audio format it introduced. MPEG-1 has been followed by the non-obviously numbered MPEG-4 part 2 [67], in 1998, also called MPEG-4 visual because of its object-oriented approach.

Interestingly, in the 90's, two lines of standards were coexisting. The ITU-T H.26X line was designed for video telephony, while the ISO/IEC MPEG was meant for digital TV broadcasting. However, both were sharing many technological aspects. There is even a certain degree of compatibility between MPEG-4 visual and H.263. Quite logically, ISO/IEC MPEG and ITU-T VCEG have been joining their effort in the development and publication of common video compression standard, thus starting a particularly successful line of video codecs.

MPEG-2/H.262 [68] has been a tremendous success in the 90's, and the enabler of widespread digital TV. MPEG-2 has been present on cable TV, satellite TV, DVD, and is still running nowadays. In the early 2000's, AVC/H.264 [69] has been a key component of the HD TV development, on traditional networks as well as on internet and mobile networks. AVC/H.264 is also used in HD Blu-Ray discs. Ten years later, in the 2010's, HEVC (H.265) [70] has been the enabler of 4k/UHD, HDR and WCG. Finally, VVC (H.266) [71] has been issued in 2020. Although it is a young codec, not yet widely deployed, it is perceived as an enabler for 8k [9] and as a strong support for the ever-growing demand for high quality video over the internet.

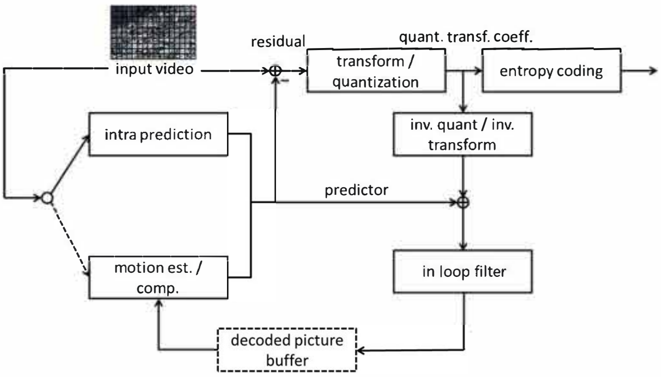

Fig. 1. The block-based hybrid video coding scheme.

B. *The block-based hybrid video coding scheme*

The block-based hybrid video coding model is depicted on Fig. *1*. It constitutes the basis of all current video compression standards. The main elements are

- Intra prediction, for coding intra frames, i.e., frames without temporal dependency, or intra blocks inside inter frames, for managing local areas that cannot be temporally predicted, such as uncovering areas.
- Inter prediction combines the capabilities of keeping a buffer of previously encoded frames and addressing these previous frames with motion compensation for efficient prediction.
- Transform and quantization are applied on residual blocks of pixels output by the prediction step. The transform tends to compact the information on a few coefficients, while the quantization adjusts the trade-off between quality and bitrate. Quantization is a lossy process.
- Entropy coding is a fundamental information theory concept. Its goal is to determine the statistically shortest representation of the data. It is a lossless process.
- In-loop filtering is applied on the frames which are stored for future temporal prediction, to improve their quality, hence the quality of the upcoming predictions. The most advanced codec, VVC, implements four successive loop filters (LF), luma mapping with chroma scaling

(LMCS), deblocking filter, sample adaptive offset (SAO), and adaptive loop filter (ALF).

Interestingly, the technology does not change radically between codec generations. Instead, the same principles and ideas are re-used and pushed further. Of course, there are new coding tools, but the overall structure remains the same.

Compared to MPEG-2, AVC/H.264 brought notably reduced complexity integer discrete cosine transform, multiple reference inter-frame prediction, in-loop deblocking filter, variable block sizes and flexible handling of interlaced video, all contributing to its excellent coding efficiency. The profiles definition allows adapting to multiple use-cases, making it suitable for any application. At the same period, the video compression research community has also been focusing on the concept of 3D wavelet filtering [10]. The wavelet transform has been used successfully in the JPEG2000 image compression standard [11]. The wavelet transform is a signal decomposition and analysis tool. Applied on an image, it replaces usual transforms such as DCT for energy compaction and provides a resolution scalable representation. That is, a wavelet compressed image can be reconstructed progressively, from lowest to highest frequencies, without coding efficiency loss. When applied to video, the same principle is extended on a 3D pixel volume [12]. The MC-EZBC codec is a good example of state-of-the-art 3D wavelet-based video coding [13]. This kind of technology was promising, but never reached the AVC/H.264 performance [14].

Jumping to the next generation, HEVC brought many improvements over AVC/H.264, including a much more flexible partitioning scheme, with up to 64x64 pixels partition sizes instead of 16x16, multiple transforms, improved motion compensation filtering and a new loop filtering restoration tool called Sample Adaptive Offset (SAO).

In parallel, a strong research focus was set on sparse modeling for image and video representation and analysis. As explained in [15], sparse coding consists in representing data with linear combinations of a few dictionary elements. Generally, the dictionary must be learned to be best adapted to the data. Considering images, the underlying idea is that only a tiny subset of the huge set of all possible pixel values combinations actually represents viewable images. Therefore, images can be represented by a smaller set of variables, as few as possible if compact representation is desired. Although the idea seems quite simple, building a dictionary is a non-trivial task. In [16], a method is proposed to learn basic texture elements representations and is used for intra coding. Video compression is tackled in [17], where dictionary learning is performed in the DCT domain of a block motion compensated structure. These methods can outperform the state-of-the-art codecs, well, if one considers AVC/H.264 as such. None of these methods reached the general performance and flexibility of HEVC. One may note though that sparse coding shines in specialized applications, such as very low bit rate human face coding [15]. Also the dictionary learning strategy anticipates the upcoming machine learning.

Finally, VVC outperforms HEVC thanks to further enhanced coding tools, such as an even more flexible partitioning scheme or a new in-loop restoration filter. Moreover, VVC includes from start several features that makes it “versatile”, including 360° video coding, screen content coding, gradual decoder refresh for low delay applications, and scalability, based on reference picture resampling (RPR) the ability to perform temporal prediction on reference images of different resolutions.

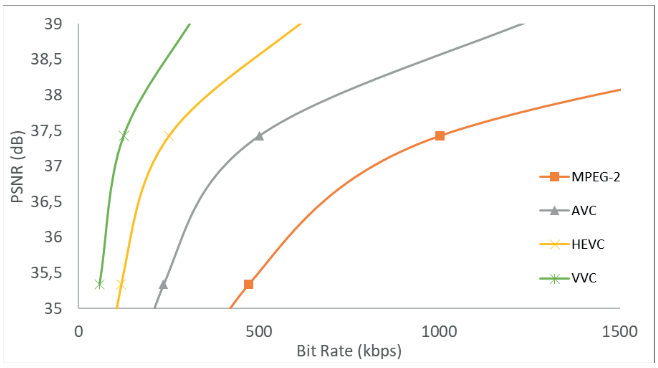

Fig. 2. Video codecs rate distortion performance progression example.

C. Video CODECs performance, limits, and discussion

Each codec generation allows decreasing the bitrate approximately by a factor two (Fig. 2). This comes however at the cost of increased complexity. For instance, the reference VVC encoder is about 10 times more complex than the reference HEVC encoder. Let us illustrate this process with a simple example: Intra prediction mode, illustrated on *Fig. 3,* which consists in encoding a block of a frame independently from previous frames. In MPEG-2, intra block coding is performed without prediction from neighboring blocks. In AVC/H.264, intra blocks are predicted from neighboring blocks, with 9 possible modes. In HEVC, the prediction principle is reconducted, with 35 possible modes, while VVC is pushing further to 67 possible prediction modes. Having more prediction modes allows better predictions, hence better compression (even though mode signaling cost increases), at the cost of more complexity.

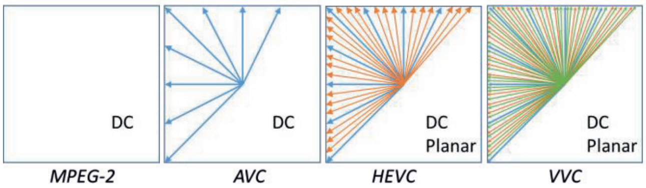

Fig. 3. Evolution of intra prediction in video codecs from MPEG-2 to VVC.

One natural question which arises is how far we can push this model. In other words, can we improve steadily the compression performance of this model decades after decades, by pushing the parameters and adding more local coding tools, or are we converging to a limit? At each codec generation, the question has been raised, and answered by the next generation. None of the proposed competing models have ever succeeded in outperforming the hybrid block-based model.

Nowadays, the recognized industry benchmark in terms of video compression performance is VVC. Can we go beyond the VVC performance? Well, the answer is already known, and it is yes. Indeed, the JVET standardization group is currently conducting explorations. The Ad-Hoc Group 12 (AHG12) is dedicated to the enhancement of VVC. Around 15% coding efficiency gains are already achieved, only two years after VVC finalization [18]. So, we may continue the process for at least another decade.

However, there is a new contender arising: artificial intelligence; or more precisely, machine learning, or deep learning. In another Ad-Hoc Group, AHG11, JVET is exploring how machine learning can be the basis of new coding tools. This also brings coding efficiency gains of about 12% [19]. Hence the question: will the future of video compression include machine learning? At this stage, we would like to point-out two new facts.

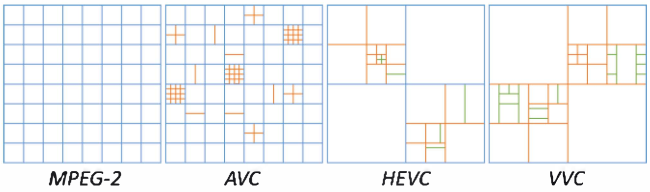

Fig. 4. Evolution of frame partitioning in video codecs from MPEG-2 to VVC.

First, considering the “traditional” methods explored by AHG12, there is a coding tool which seems to stop bringing gains: frame partitioning. The partitioning is a fundamental tool for video compression. It defines how precise can be the adaptation of the encoder to local content characteristics. The more flexible it is, the better the coding efficiency. All the subsequent coding tools depend on the ability to partition the frame efficiently. As illustrated on *Fig. 4,* AVC/H.264 has 16x16 pixels blocks, with some limited sub-partitioning. HEVC implements a much more flexible quadtree based partitioning from 64x64 pixels blocks. VVC combines quadtree partitioning with binary and ternary tree partitioning, from 128x128 pixels blocks for even more flexibility. During the exploration following HEVC standardization, the single fact of enhancing the partitioning brought up to 15% coding efficiency gains. Similarly, in the AHG12 context, people came with new extended partitioning strategies. However, only marginal gains were reported [20]. Does that mean we are finally approaching a limit?

The second fact is the development of end-to-end deep learning video compression. This strategy is highly disruptive. In short, the whole block-based hybrid coding scheme is replaced by a set of deep learning networks, such as auto-encoders. These types of schemes are competing with state-of-the-art fixed image coders [21]. For video applications, they are matching HEVC performance [22][23]. This level of performance has been reached in only five years. That’s an unprecedently fast progression. One may easily extrapolate, even if the progression slows down, that the state-of-the-art video compression performance will soon be the end-to-end strategy prerogative. Therefore, we may very well be at a turning point of the video codecs history.

III. THE ADVENT OF ARTIFICIAL INTELLIGENCE

Artificial intelligence (AI) refers to technologies that allow computers to perform tasks that have so far required human intelligence. The AI term is born in the 1950s with among others the work of Alan Turing and its famous Turing test [34]. Already at this time, researchers tried to create artificial neurons to mimic the human brain. The first neural machine, the Stochastic Neural Analog Reinforcement Calculator (SNARC) has been built in 1951 by Marvin Minsky and has been the beginning of larger research in this field. This leads to the creation of the well-known Perceptron in 1957 which is the basis of modern deep learning. But researchers were too optimistic creating an intelligent machine. After several failures, funding and interest in the field dropped off, leading to the first AI winter. Researchers were mainly constrained by the limited computing power. Some of them persisted in the idea of creating a machine capable of carrying out complex human tasks, and in 1997, IBM’s Deep Blue became the first computer to beat a chess champion. A lot of modern deep learning architectures such as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) were designed during the 1980s et 1990s, but they required too many data and power to train, and they were forgot during several years. Researchers then focused on more practical and more humble problems. This was the emergence of machine learning (ML).

ML is a technology that allows algorithms to realize tasks without having been explicitly programmed to do them. It is a subfield of artificial intelligence that let the algorithms discover patterns in data to improve their performances on a specific task. A dataset needs to be prepared in order to train the model for a given task. This is the training step. After several iterations, the algorithm, or model, learns to extract more useful features from the dataset which makes him do better predictions. The model is then evaluated on a validation set to check that it generalizes well, i.e., that it performs as well on data that it has never seen before. ML have been used in different fields such as speech or character recognition for example, and many of the ML algorithms widely used today were invented before the 2000s (nearest neighbor, boosting, multilayer perceptron, …)

In the 2000s, computing power had improved, and the era of big data started to make a lot of, well, data available. AI started to succeed in many industrial use cases such as robotic. In the 2010s, computing power had improved further, specifically with the progress of Graphical Processing Units (GPUs). Moreover, public datasets were built with annotated samples, such as ImageNet [35], created for image classification task. Few years after the dataset creation, ImageNet launched the ImageNet Large Scale Visual Recognition Challenge (ILSRVC), an annual AI object recognition challenge. This led to the first deep learning-based solution in 2012 [36], outperforming all other solutions based on traditional machine learning.

This work triggered an explosion of applications using deep learning technologies, allowing to achieved performances never reached before on various tasks related to natural language processing and computer vision. Deep learning allows to automatically create a hierarchical representation of data, highlighting features and their relations that are hard, if possible, to describe manually. Today, DL allows cars to drive themselves, robots to communicate with humans, but it can also generate data, with automatic picture colorization or realistic/artistic creations such as the recent Meta's make-a-scene tool [32]. This field is growing very quickly since a decade, and models improve every year.

IV. Machine learning based video compression

A. Tool based ML-based video compression

As AI and ML is gaining attention in all possible fields of research, video compression is no exception. The JVET standardization group has started an exploration activity dedicated to the introduction of ML in the VVC framework [38]. The idea here is to keep the hybrid block-based model and to replace or complement elementary coding tools by ML-based tools.

Intra prediction is addressed in [39] and [40]. Both approaches consider the prediction of a block of pixels from a causal pixel neighborhood. The prediction is performed by a neural network (NN) replacing the traditional directional or planar intra prediction (*Fig. 3*). The NN is expected to be able to predict complex shapes and textures. 2 to 3% coding efficiency gains are reported.

Inter prediction is considered using several strategies. In [41], an enhanced bi-prediction mode is proposed. Instead of predicting a block with an average of two motion compensated reference blocks, the two predictors are fed to a NN which outputs the final prediction. Up to 1% coding efficiency gains are reported. A similar approach is considered in [42], but with a single predictor. In [43], a whole reference frame is generated by a NN. This new frame is added to the reference list for temporal prediction. Therefore, each image block can be predicted either from a past encoded frame or from a NN generated frame without changing fundamentally the encoding/decoding process. Up to 2% coding efficiency gains are reported.

A strong focus has also been set on loop filtering [38]. The general idea of loop filtering is to restore already encoded frames. It serves both as a post processing, improving the visual quality of the compressed video, and as a coding efficiency improvement, as it allows better temporal predictions. As ML has been explored for many image processing tasks, it is a natural candidate for loop filtering. First attempts considered replacing all the loop filters by a ML process. The idea has then been refined with adaptive methods trying to take advantage of the best of two worlds, signal processing and ML. The CNNLF [44] is proposed as an alternative to the deblocking and SAO filters of VVC. It is up to the encoder to decide locally to activate CNNLF or not. The filter inputs many data, including quantization parameter (QP), a prediction image and a partition image. The filter also includes a scaling as a post processing after the NN step. Up to 12% coding efficiency gains are reported, illustrating the huge impact of the single loop filter coding tool. The filter proposed in [45], replaces all VVC LF, though it can be turned off at block level. It similarly relies on rich input and post-scaling, while making use of attention models. Similar performance is reported.

Overall, the coding efficiency gains obtained with these approaches are largely significant. However, they come at an unprecedented cost in complexity, with figures going up to 400 times slow-down of the decoder.

B. Super-resolution-based video compression

The idea of using super-resolution stems from the well-known over the top (OTT) streaming concept. Depending on available bitrate there exists an optimal combination of resolution and compression tuning. In other words, when the bitrate decreases, it becomes more efficient to decrease the video resolution rather than getting more compression artifacts. By factoring in the fact that ML has been studied as a mean of recovering fine details when increasing image resolution, one arrives naturally to the idea that encoding videos at a lower resolution may lead to better trade-off than the current approaches, thanks to the capability of NN to up-sample content without generating the traditional blurring and aliasing phenomenon.

In [46], the video sequence is first decomposed before being compressed using a traditional codec. It is then synthesized to retrieve the original resolution. The decomposition consists in down-sampling only the inter coded frames. Thus, the intra frames are carrying texture information, while inter frames are carrying the temporal information. Synthesis, or up-sampling, is assisted by a motion compensated NN. Up to 9% coding efficiency gains are reported.

In [47], the whole video is encoded at a lower resolution. NN-based super-resolution is applied as a post-processing to recover the original resolution. However, in order to better adapt to frame characteristics, the last layer of the NN is specialized for each sequence. The corresponding parameters are transmitted along with the video stream. About 6.5% coding gains are reported.

Depending on the complexity of the chosen NN approach, complexity gains can be observed. Indeed, thanks to the lower video resolution, the coding/decoding step is much less complex, potentially compensating for the NN complexity [48].

Finally, [49] proposes a different approach. No NN based super-resolution is involved. Rather, linear down/up-sampling is performed using the VVC RPR feature. The point is that resolution is chosen on encoder side by classical rate-distortion optimization. Therefore, depending on the sequence, one obtains either identical or better results than VVC. It shows that ML is not necessary to obtain gains by playing with resolution.

C. End -to-end learned video compression

Nowadays, learned image and video compression has been the target theme for both the Machine Learning and image/video compression communities. In this context, the Challenge on Learned Image Compression (CLIC) [37] aims to encourage both communities to advance the field of image and video compression using machine learning algorithms by either designing new codec architectures or by introducing new perceptual metrics. Therefore, each year, publications are gathered, evaluated, and compared against the traditional codecs. Then, the winners are presented in the CVPR workshop.

Recently, learned image compression has achieved significant progress in coding performance. The state-of-the-art of such methods are currently competitive with the latest traditional coding system VVC in intra mode. Inspired by this success, deep learning methods were extended to learned video coding.

Learned video compression approaches can be divided into two main categories. The first one keeps the traditional coding pipeline, that deals with the inter frame redundancies, unchanged (motion estimation, motion compensation, residual coding). Then, for each step, deep learning architectures such as auto-encoders and optical flow architectures are used. For instance, [51] introduce the first low latency compression framework called DVC using auto-encoders to code motion vectors and residuals, a pretrained optical flow model for motion estimation and a bilinear warping for motion compensation. [53] improves the DVC performance by using multiples frames as references. This new coding system is called MLVC, it added four neural modules to the DVC framework. The first explored a buffer of multiple previous motion vectors to achieve motion estimation of the current frame. The second does the same for motion compensation. The two remaining modules aim to refine the motion vectors and the residuals, respectively. In the same context of low latency coding, [55] introduces a recurrent learned video codec (RLVC) using a recurrent autoencoder and a recurrent probability model to compress the motion and the residual features. The goal is to thoroughly explore the temporal correlation between frames and latent representations. In fact, this work enables using all the previous decoded frames as reference for compressing the current frame. In addition, the recurrent probability model tends to achieve lower bitrate since the latent representation of the current frame is conditioned with the previous ones. While DVC manages to outperform the low-delay P frames (LDP) configuration of x264 and compete with the same configuration of x265, MLVC and RLVC outperform DVC and x265 in terms of coding efficiency.

[52] presents a framework to code a GOP structure, which includes P and B frames, with different level of quality (HLVC). P and B frames are coded using two architectures of networks which achieve motion estimation, motion compensation and residual coding with hierarchical quality levels. Then, a Recurrent Neural Network (RNN) module is used to enhance quality of the decoded frames. This work's proposed framework depends on GOP structure, which is set manually before proceeding to the training stage. Although this method codes a GOP structure with B and P frames, it was evaluated against the LDP mode of x265 and the low latency model DVC. Compared with x265, it manages to obtain gains in BDBR: -6% for PSNR models and −35.94% for MS-SSIM models. [54] presents a method to achieve perceptual learned video compression (PLVC) using a recurrent conditional GAN. This framework consists of a compression network based on RLVC [55], that serves as generator, in addition to a recurrent discriminator that take as input spatial and temporal conditions as well as the current and previous reconstructed frames. The training process minimize a combination of an adversarial loss function with the rate distortion one. This work manages to get the best results in terms perceptual metrics such as LPIPS [60] and FID [61] compared to the leaned codecs: RLVC, HLVC, MLVC and the traditional codec HEVC (HM16.20). However, in terms of objective metrics like PSNR and MS-SSIM, it is on-par with DVC, and it performs worse than the previously mentioned learned video codecs as well as HEVC (HM 16.20).

While in the previous works, I frames are compressed using either the BPG codec for [51][53][55][52] or a learned image codec for[54], [56] proposes a neural coding framework for I and P frames, and [22] introduced a neural architecture, consistent with all type of frame I, P and B frames. The system contains two networks: MOFNET deals whith motion estimation and compensation and CodecNet achieves conditional coding which replace residual coding. This approach achieves performance competitive with the state-of-the-art video codec HEVC (HM 16.20).

The second category of end-to-end learned video compression focuses on reducing temporal redundancy using algorithms that are different from the traditional pipeline. For example, [57] proposes a video compression framework based on an 3D auto-encoder combined with temporally conditioned entropy model. The performance of this method is competitive with x265 in terms of MS-SSIM. Other works used frame interpolation for video coding. [58] explores Generative Adversarial Networks (GAN) as a decoder to reconstruct separate frames, then used linear interpolation to reconstruct the missing frames. This work is evaluated on low resolution gray sequences in low bitrate. Unfortunately, the results of this approach are only comparable with MPEG4. [59] uses a learned image codec to compress key frames and then uses an interpolation model to predict the missing frames. This approach is compared with handcrafted codecs such as HEVC, AVC and MPEG on the VTL dataset [62]. It outperforms MPEG4 and is matching H264.

All in all, although learned video coding in intra mode (learned image coding) performance is on-par with the latest handcrafted codec VVC, extracting spatiotemporal features is more challenging which makes learned inter coding more difficult. Therefore, the state-of-the-art of learned video compression currently matches the coding efficiency of HEVC. However, one can predict that the progress in this field will be significant in a short period of time.

V. Practical application of machine learning based video compression

A. *Delay, rate-control and content adaptation*

There is a huge difference between a codec, as defined by standards, and a ready to production live video encoder. The codec is only a part of a video encoder. A video encoder must manage various inputs or capture, decoding, encoding, muxing and output, all along with system functions and user interface. Even when focusing on the encoding part, there is more than the codec. Live encoding requires optimization of the complexity/quality trade-off, which generally translates into added delay. This delay must of course stay under control. Delay is caused among other things by Pre-processing and analysis in a look-ahead buffer, frame reordering for efficient group of pictures (GOP) structure coding, pipelining, and rate-control.

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

Content adaptation is desirable for optimal quality. GOP structure is generally adapted to the nature of the content. Scene-cuts are detected, and temporal prediction is avoided across them. Considering end-to-end video coding, the same ideas may apply. However, depending on the end-to-end implementation, it may be simpler. The idea of GOP structure may be managed in a transparent manner by the ML model. The notion of successive GOP may be easily conserved, allowing easy chunking for OTT and short zapping time.

In short, content adaptation does not seem to be an obstacle to the end-to-end video encoders development. Rate-control, on the other hand, may be more difficult. Indeed, in traditional video coding, there is an understandable, though non-trivial, relationship between the QP and the bitrate. In an end-to-end video encoder, there exist a parameter tuning the bitrate. However, the effect of this parameter is generally not easy to model. Some encoders are actually trained for a single value of this parameter. It implies that if one needs 64 rate levels, like the 64 QP values of VVC, 64 models must be trained and stored. In an attempt to answer to this issue, [50] proposes a new loss function, where the λ parameter, responsible for rate tuning, is non constant. It allows to design a training procedure where several values of λ are fed randomly to the system, thus making a model that can react appropriately to any value of λ at inference time. Literature on this topic is limited as of today, and there is no doubt that further research is needed, but this example is encouraging.

Finally, the main difficulty to handle may very much be the huge operational complexity of NN.

B. Computing resources

During several years after the deep learning emergence, researchers did not really care about models' complexity. The solutions proposed for various public challenges were more complex every year, while their performances continued to grow exponentially. In their analysis, [24] have shown that the largest model training runs have doubled the computational power used every 3.4 months since 2012. As an example, Danish researchers used the "Carbontracker" tool [25] to show that the energy required to train a GPT-3 model (one of state-of-the-art model for natural language tasks) could have the carbon footprint of driving 700,000km. The training step of machine learning models is highly resource-intensive, but the inference one consumes far more power. Indeed, while the model is trained once, it can be used for billions of inferences. It is estimated that inference accounts for up to 90% of the computing cost [26].

The increase in the model's complexity has been made possible thank to the hardware evolution. For deep learning technologies, Graphical Processing Units (GPUs) are often the default choice, because of their ability to perform a lot of low-level mathematical operations in parallel. Initially designed for games and graphically intensive applications, researchers thought their capabilities were suited to run deep learning models. This market is dominated by Nvidia, and since the deep learning development, they have built new GPU architectures that make their hardware more effective for models training and inference. But this kind of hardware still is a general-purpose solution. Some manufacturers decided to build specific chips designed to run deep learning models even more effectively. One can think about Google Tensor Processing Units (TPUs), or Microsoft Catapult project. They are based respectively on Application-Specific Integrated Circuits (ASIC) and Field Programmable Gate Array (FPGA) and allow power consumption reduction related to GPUs. These solutions are available in cloud infrastructures, so they can be used for models training and online inference. These use cases are rarely constrained by consumption resources. If more power is needed to speed up training or inference, it is simple to scale by adding GPUs for example. But what about edge devices?

Edge devices are appliances on which data collection takes place. It can be desktop computers, smartphones, or connected devices. While GPUs or TPUs are still the default solutions for training models, a lot of works has been done for performing inference on edge devices. In contrary to cloud platforms, scaling is very hard due to limits in space, power, and connectivity. But this is a very important use case as it allows processing data locally, mitigating networks limitations, increasing security, and improving data privacy. Researchers and manufacturers have then put a lot of effort improving edge computing hardware for processing deep learning models. Hence new types of AI-optimized accelerators have been designed during the past few years, that can be regrouped under the name Neural Processing Units (NPUs). Main mobile manufacturers have designed their own solution. This includes chips such as the Apple Neural Engine, the Kirin 980 from Huawei, or the Exynos 9820 from Samsung. There also exists development boards such as the Nvidia Jetson Nano or the Google Coral Edge TPU. NPUs are based on specific architectures that make deep learning model execution faster while having limited consumption. A lot of accelerators exist today [27], and this is a very active research field. Few years ago, MLPerf benchmarks [28] have been released in order to make AI platforms performances comparison simpler. It allows to get training time, inference time, and more recently power consumption of a specific hardware configuration for different AI models. Despite these initiatives, AI accelerators comparison remains very hard as performances are related to too many factors, not only the accelerator itself. Performances are also impacted by the CPU, and the software library used to deploy the model.

In addition to work on specialized hardware, a lot of work has been done on the software part. Some of them are designed for CPUs (OpenBLAS, Intel MKL, …), and others for GPUs (cuBLAS, cuDNN, …). All of them optimize matrix operations in order to make AI model execution faster using only algorithmic optimizations. These are libraries allowing low-level mathematic operations, but they are mainly used through higher-level frameworks and tools. For example, Openvino [29] and TensorRT [30], respectively developed by Intel and Nvidia, are platforms offering runtimes with optimized operations implementation, but also some model optimization strategies. This includes weights quantification, network pruning, or operations fusion.

The combination of hardware and software evolution allows the execution of powerful AI models on edge devices in real time. But the AI field is evolving really fast. Even with this progress, models' complexity keeps growing every year,

http://www.set.org.br/ijbe/ doi: 10.18580/setijbe.2022.3 Web Link: http://dx.doi.org/10.18580/setijbe.2022.3

and hardware and software providers must continue to improve their solutions to make model execution faster or less energy consuming. Recent trends such as neuromorphic computing [31] show there is room for improvement with completely different designs. Also, new hardware is challenging dominant existing solutions. For example, the Hailo 8 chip [33] presents performances up to 13x those of Google TPUs. All of this shows that the Moore's law continues and makes possible further improvements in AI.

VI. A Case-study: end-to-end memory consumption

A. Problem statement

As models' sizes are growing continuously, memory consumption is also becoming an issue, along with computing power. The case of end-to-end learned encoding is considered here. The auto-encoder architecture, built with convolutional layers, enables processing different video resolutions, no matter the resolution used during the training step. However, with growing models' sizes and video resolutions (4K, 8K), these solutions are facing hardware memory saturation. One way to solve this issue is to use a patch-based coding approach. The video frames are divided into patches smaller than the frame size, that can be encoded independently. Then, the decoded patches are gathered to reconstruct the decoded frames.

This solution addresses the hardware limitation issues, but the reconstructed frames can have block artifacts at the patch boundaries, widely deteriorating the video quality.

B. Patch-based end-to-end video encoding

A solution to the memory saturation is proposed to perform patch encoding while removing block artifacts. The idea is to encode overlapping patches and then use a linear function to combine the reconstructed overlapped pixels. If b_m and b_{m+1} are two consecutive reconstructed patches overlapping horizontally on N pixels, the value of the i^{th} overlapped pixel $p_{rec}(i)$ for a given line in the reconstructed frame is determined by the following equation:

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

where $i \in \{0, \ldots, N-1\}$ is the index of the overlapped pixels, P is the size of the patch without overlapping, p_{b_m} and $p_{b_{m+1}}$ are pixels values, for a specific line, of two consecutive decoded patches b_m and b_{m+1} respectively. The same equation applies for vertically overlapping patches.

The proposed approach has been applied to encode I frames, using an end-to-end learned image codec which is an implementation of the model architecture introduced in [21]. This model was trained on CLIC 2020 dataset [37]. For training, 256×256 sized patches were randomly cropped from each image of the training set. The loss function to be minimized is:

$$J = D + \lambda R \qquad (2)$$

where D refers to the distortion measured by the Mean Square Error (MSE) or the Multi-Scale Structural Similarity Index (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 method is then evaluated on Class B, C, D, E and F of the JVET Common Test Conditions (CTC) sequences (8-bit sequences) [72]. For each sequence, one frame is extracted and compressed both entirely (referred to as the full image approach) and by the proposed patch-based approach, with and without overlapping, where $N \in \{0, 2, 4, 8, 16, 32\}$ overlapped pixels and $P = 256$, as the training resolution.

BD-rate gains of the patch-based learned image coding with and without overlapping were computed comparing to full image learned image coding, using an end-to-end model trained to minimize MSE as distortion metric.

For MSE models, patch-based image coding without overlapping presents a slight loss in BD-rate (Average BD-rate +0.013), comparing to full image coding, which mostly corresponds to the block artifacts issue caused by patch-based approaches.

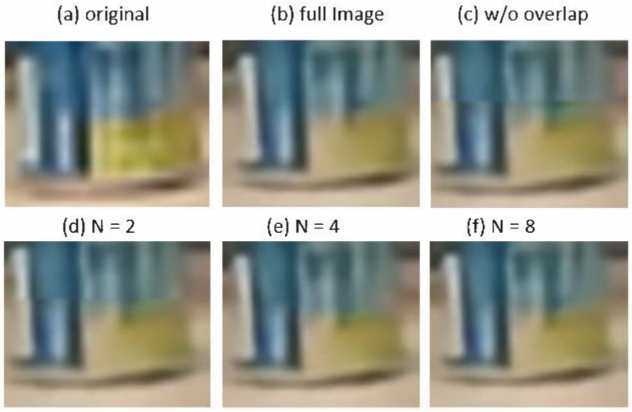
(a) original (b) full Image (c) w/o overlap
(d) N = 2 (e) N = 4 (f) N = 8

Fig. 5. Visual results of comparison for FourPeople. The model used optimizes the MSE metric with λ = 4096.

On the other hand, the proposed method achieves a gain in BD-rate, which increases as the number of overlapped pixels is increased. For N = 2, the average BD-rate gain among CTC sequences is: -0.025, which shows that two overlapped pixels seems to be sufficient to eliminate the borders artifacts. With N = 8 and N = 16, small gains are observed comparing to full image coding: -0.034 and -0.041 respectively. For N > 16, BD-rate gains saturation is observed. An example of decoded images is presented in Fig. 5. Overlapping with N = 2 and N = 4 reduce the block artifacts while overlapping with 8 pixels eliminates them entirely.

The experimental complexity and memory consumption are reported in Table I. Frames of different resolutions were extracted from the JVET CTC and were coded using two machines with powerful GPUs: GeForce RTX 2080ti and 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. It is important to note that these resolutions are standard resolutions in practical applications of image compression. 4k is not even considered. Therefore, the fact that they cannot be run on one of the latest GPUs is inconvenient. In this case, the proposed method provides a solution that enables coding high resolution images without deteriorating quality. While the method is necessary for resolution 720p and above, it is adding some 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 to overlap patches.

To conclude this section, the proposed approach addresses the hardware memory limitation problem since it allows coding resolutions such as HD and 720p, while maintaining same or better quality as the full resolution learned coding.

TABLE I: PERFORMANCE OF PATCH BASED END-TO-END ENCODING.

Resolution	Method	Coding Time GPU 2080 11Go	Coding Time GPU 3090 24Go
1920x1080	Full Resolution Coding	OOM	OOM
	Patch coding in parallel with overlapping	3.82s	2.05s
1280x720	Full Resolution Coding	OOM	0.93s
	Patch coding in parallel with overlapping	1.91s	1.012s
832x480	Full Resolution Coding	1.06s	0.52s
	Patch coding in parallel with overlapping	1.10s	0.55s

VII. WRAP-UP

From MPEG-2 in the 90's to VVC nowadays, four successive major generations of codecs have made video ubiquitous, from TV screen to smartphones, from over-the-air to internet. All these codecs are based on the same general structure, the hybrid block-based model. Previous attempts to overcome this model have all failed, despite of their numerous technical qualities and features. But how long will this model continue to dominate?

Today, one observes a small hint of a decline of the hybrid block-based model, along with the rise of machine learning. Machine learning is the state-of-the-art technology in many image and video processing fields, but still not in video compression. ML may not be ready yet for video compression, but it is progressing fast. We argue in this paper that current limitations can be addressed, either through plain technological progress, or through dedicated algorithmic progress.

As an example, a new method of memory management for machine learning based end-to-end image and video compression is described in this paper, namely patch encoding with overlapping.

All in all, for the upcoming video codec generation, two approaches are competing. Time will tell, but our guess is that there will be another generation of hybrid block-based model before the advent of machine learning based video compression. Researchers are just needing a few years to refine and make the technology practical. Model sizes and hardware capabilities will eventually converge.

REFERENCES

[1] Cisco Annual Internet Report (2018–2023) White Paper, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html

[2] Video Encoder Market with COVID-19 Impact by Number of Channel (Single, Multichannel), Mounting Type (Standalone, Rack-mounted), Application (Broadcast, Surveillance (Commercial, Residential, Institutional)), and Geography - Global Forecast to 2025, https://www.marketsandmarkets.com/Market-Reports/video-encoder-market-109133493.html.

[3] R. D. Kell. Improvements relating to electric picture transmission systems. British Patent No. 341,811, 1929.

[4] Thomas Wiegand and Heiko Schwarz (2016), "Video Coding: Part II of Fundamentals of Source and Video Coding", Foundations and Trends® in Signal Processing: Vol. 10: No. 1–3, pp 1-346. http://dx.doi.org/10.1561/2000000078.

[5] Gary Sullivan, "Overview of International Video Coding Standards (preceding H.264/AVC)" ITU-T VICA Workshop, 22-23 July 2005, ITU Headquarter, Geneva.

[6] S.Vetrivel et. al., "An Overview Of H.26x Series And Its Applications", International Journal of Engineering Science and Technology, Vol. 2(9), 2010, 4622-4631

[7] https://www.itu.int/

[8] https://www.iso.org/about-us.html

[9] T. Biatek, M. Abdoli, T. Guionnet, A. Nasrallah and M. Raulet, "Future MPEG standards VVC and EVC: 8K broadcast enabler" IBC365, 14 September 2020.

[10] E. Moyano, F. J. Quiles, A. Garrido, T. Orozco-Barbosa and J. Duato, "Efficient 3D wavelet transform decomposition for video compression," Proceedings Second International Workshop on Digital and Computational Video, 2001, pp. 118-125, doi: 10.1109/DCV.2001.929950.

[11] Taubman, D., Marcellin, M. (2012). *JPEG2000 Image Compression Fundamentals, Standards and Practice: Image Compression Fundamentals, Standards and Practice* (Vol. 642). Springer Science & Business Media.

[12] V. Bottreau, M. Benetiere, B. Felts and B. Pesquet-Popescu, "A fully scalable 3D subband video codec," *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, 2001, pp. 1017-1020 vol.2, doi: 10.1109/ICIP.2001.958669.

[13] Peisong Chen and J. W. Woods, "Bidirectional MC-EZBC with lifting implementation," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 10, pp. 1183-1194, Oct. 2004, doi: 10.1109/TCSVT.2004.833165.

[14] P. Lambert, W. De Neve, P. De Neve, I. Moerman, P. Demeester and R. Van de Walle, "Rate-distortion performance of H.264/AVC compared to state-of-the-art video codecs," in *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 1, pp. 134-140, Jan. 2006, doi: 10.1109/TCSVT.2005.857783.

[15] Julien Mairal; Francis Bach; Jean Ponce, *Sparse Modeling for Image and Vision Processing* , now, 2014.

[16] Y. Sun, M. Xu, X. Tao and J. Lu, "Online dictionary learning based intra-frame video coding via sparse representation," *The 15th International Symposium on Wireless Personal Multimedia Communications*, 2012, pp. 16-20.

[17] Irannejad, M., Mahdavi-Nasab, H. Block Matching Video Compression Based on Sparse Representation and Dictionary Learning. *Circuits Syst Signal Process* **37,** 3537–3557 (2018). https://doi.org/10.1007/s00034-017-0720-5.

[18] JVET-Z0012-v1 "JVET AHG report: Enhanced compression beyond VVC capability (AHG12)" Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29 26th Meeting, by teleconference, 20–29 April 2022.

[19] JVET-Z0023 "EE1: Summary of Exploration Experiments on Neural Network-based Video Coding" Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29 26th Meeting, by teleconference, 20–29 April 2022.

[20] JVET-Y0150-v2 "EE2-1.1: Tests on unsymmetric partitioning methods" Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29 25th Meeting, by teleconference, 12–21 January 2022.

[21] Z . Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 7936–7945, 2020

[22] Ladune, T., Philippe, P., Hamidouche, W., Zhang, L., & Déforges, O. (2021). Conditional Coding for Flexible Learned Video Compression. 1–18. http://arxiv.org/abs/2104.07930

[23] Hu, Zhihao and Lu, Guo and Xu, Dong, "FVC: A New Framework Towards Deep Video Compression in Feature Space", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2021,pp 1502-1511.

[24] Amodei, D. & Hernandez, D. (2018). AI and Compute. https://openai.com/ blog/ai-and-compute/. 3, 12, 14

[25] Anthony, L. F. W., Kanding, B., & Selvan, R. (2020). Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. arXiv preprint arXiv:2007.03051.

[26] Thomas D.. Reducing machine learning inference cost for pytorch models - aws online tech talks. https://www.youtube.com/watch?v=ET2KVe2du3Y, 2020.

[27] Reuther, A., Michaleas, P., Jones, M., Gadepally, V., Samsi, S., & Kepner, J. (2021, September). AI accelerator survey and trends. In 2021 IEEE High Performance Extreme Computing Conference (HPEC) (pp. 1-9). IEEE.

[28] Reddi, V. J., Cheng, C., Kanter, D., Mattson, P., Schmuelling, G., Wu, C. J., ... & Zhou, Y. (2020, May). Mlperf inference benchmark. In 2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA) (pp. 446-459). IEEE.

[29] https://docs.openvino.ai/latest/index.html#

[30] https://developer.nvidia.com/tensorrt

[31] Imam, N., & Cleland, T. A. (2020). Rapid online learning and robust recall in a neuromorphic olfactory circuit. Nature Machine Intelligence, 2(3), 181-191.

[32] Gafni, O., Polyak, A., Ashual, O., Sheynin, S., Parikh, D., & Taigman, Y. (2022). Make-a-scene: Scene-based text-to-image generation with human priors. arXiv preprint arXiv:2203.13131.

[33] https://hailo.ai/products/hailo-8/

[34] Turing, A. M. (2009). Computing machinery and intelligence. In Parsing the turing test (pp. 23-65). Springer, Dordrecht.

[35] Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[36] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, 25.

[37] "Workshop and c. on learned image compression," Https://www.compression.cc/, 2020.

[38] Elena Alshina et al., "JVET AHG report: Neural network-based video coding", JVET-AA0011-v1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 27th Meeting, by teleconference, 13–22 July 2022.

[39] Maria Meyer and Christian Rohlfing, "AHG11-related: Investigation on CNN-based Intra Prediction", JVET-U0105-v3, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 21st Meeting, by teleconference, 6–15 Jan. 2021.

[40] T.Dumas et al.,"EE1 test 3.1: intra prediction using neural networks" ,JVET-Y0082-v2, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022

[41] F. Galpin et al., "AHG11: Deep-learning based inter prediction blending", JVET-V0076-v2, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29 22nd Meeting, by teleconference, 20–28 Apr. 2021.

[42] Changyue Ma et al., "AHG11: Neural Network Based Motion Compensation Enhancement for Video Coding", JVET-Y0090-v1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.

[43] Zizheng Liu et al., "AHG11: NN-based Reference Frame Interpolation for VVC Hierarchical Coding Structure", JVET-Y0096-v1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, 25th Meeting: by teleconference, 12–19 Jan. 2022.

[44] Liqiang Wang et al., "EE1-1.1: neural network based in-loop filter with constrained storage and low complexity", JVET-Y0078-v2, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.

[45] Yue Li et al., "EE1-1.2: Test on Deep In-Loop Filter with Adaptive Parameter Selection and Residual Scaling", JVET-Y0143-v2, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.

[46] Ming Lu et al., "EE1: Tests on Decomposition, Compression, Synthesis (DCS)-based Technology", JVET-V0149, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 22th Meeting, by teleconference, 20 – 28 Apr. 2021.

[47] Takeshi Chujoh et al., "EE1.2: Additional experimental results of NN-based super resolution (JVET-U0053)", JVET-V0073-v1, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 22nd Meeting, by teleconference, 20–28 Apr. 2021.

[48] Chaoyi Lin1 et al., "EE1-2.3: CNN-based Super Resolution for Video Coding Using Decoded Information", JVET-Y0069-v21, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.

[49] Elena Alshina et al., "EE1-2.1: Super Resolution with existing VVC functionality", JVET-Y0061-v21, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 25th Meeting, by teleconference, 12–21 January 2022.

[50] Chaoyi Lin et al., "AHG11: Variable rate end-to-end image compression", JVET-U0102, Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29, 21st Meeting, by teleconference, 6–15 Jan. 2021.

[51] Lu, Guo and Ouyang, Wanli and Xu, Dong and Zhang, Xiaoyun and Cai, Chunlei and Gao, Zhiyong, "DVC: An End-To-End Deep Video Compression Framework", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June, 2019

[52] Yang, R., Mentzer, F., Van Gool, L., & Timofte, R. (2020). Learning for Video Compression With Hierarchical Quality and Recurrent Enhancement", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2020

[53] Lin, Jianping and Liu, Dong and Li, Houqiang and Wu, Feng, "M-LVC: Multiple Frames Prediction for Learned Video Compression", Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June, 2020

[54] Ren Yang, Luc Van Gool, and Radu Timofte.."Perceptual Learned Video Compression with Recurrent Conditional GAN", arXiv preprint arXiv:2109.03082 (2021).

[55] R. Yang, F. Mentzer, L. Van Gool and R. Timofte, "Learning for Video Compression With Recurrent Auto-Encoder and Recurrent Probability Model," in IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 2, pp. 388-401, Feb. 2021,

[56] H. Liu et al., "Neural Video Coding Using Multiscale Motion Compensation and Spatiotemporal Context Model," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 31, no. 8, pp. 3182-3196, Aug. 2021

[57] Habibian, Amirhossein and Rozendaal, Ties van and Tomczak, Jakub M. and Cohen, Taco S., "Video Compression With Rate-Distortion Autoencoders", Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October, 2019

[58] S. Santurkar, D. Budden and N. Shavit, "Generative Compression," 2018 Picture Coding Symposium (PCS), 2018, pp. 258-262,

[59] Wu, Chao-Yuan and Singhal, Nayan and Krahenbuhl, Philipp, "Video Compression through Image Interpolation", Proceedings of the European Conference on Computer Vision (ECCV), September, 2018.

[60] Richard Zhang, Phillip Isola, Alexei AEfros, et al., "The unreasonable effectiveness of deep features as a perceptual metric." In CVPR, 2018.

[61] Martin Heusel, Hubert Ramsauer, et al. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In NeurIPS, 2017.

[62] Video trace library. http://trace.eas.asu.edu/yuv/index.html, 2001. Accessed: 2020-11-11. 5, 6

[63] ITU-T Recommendation H.120 (11/88).

[64] ITU-T Recommendation H.261 (03/93).

[65] ITU-T Recommendation H.263 (01/05).

[66] ISO/IEC JTC 1/SC 29 (2010-07-17). "MPEG-1 (Coding of moving pictures and associated audio for digital storage media at up to about 1,5 Mbit/s)".

[67] ISO. "ISO/IEC 14496-2:2004 - Information technology -- Coding of audio-visual objects -- Part 2: Visual"

[68] ITU-T Recommendation H.262 (02/12).

[69] ITU-T Recommendation H.264 (08/21).

[70] ITU-T Recommendation H.265 (08/21).

[71] ITU-T Recommendation H.266 (04/22).

[72] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sühring, "VTM Common Test Conditions and Software Reference Configurations for SDR Video," document JVET-T2010 of JVET, Oct 2020.

Thomas Guionnet is a Fellow Research Engineer at Ateme, where he currently leads the Innovation team's research on artificial intelligence applied to video compression. Beyond his work for Ateme, he has also contributed to the ISO/MPEG - ITU-T/VCEG – VVC, HEVC and HEVC-3D standardization process; he teaches video compression at the ESIR engineering school in Rennes; and he has authored numerous publications including patents, international conference papers, and journal papers. Prior to joining Ateme, he spent 10 years at Envivio conducting research on real-time encoding, video-preprocessing, and video quality assessment. He holds a Ph.D. from Rennes 1 university.

Marwa Tarchouli received an engineering diploma in electronic engineering from Ecole Nationale Supérieure d'Electronique, Informatique, Télécommunications, Mathématique et Mécanique de Bordeaux (ENSEIRB MATMECA), Bordeaux, France, in 2020. Since 2021, she is a PhD student at Ateme and INSA Rennes. Her Phd focuses on improving video coding schemes using machine learning algorithms.

Sébastien Pelurson received a PhD degree in computer science from the University of Grenoble Alpes, Grenoble, France, in 2016. From 2016 to 2019, he worked in the field of augmented reality, and more specifically on the use of deep learning models to improve 3D tracking algorithms on mobile devices. He joined Ateme, Rennes, France, in 2020. His research interests include video coding optimization based on machine learning technology.

Mickaël Raulet is the chief technology officer at ATEME, where he drives research and innovation with various collaborative research and development projects. He represents ATEME in several standardization bodies: ATSC, DVB, 3GPP, ISO/IEC, ITU, MPEG, DASH-IF, CMAF-IF, SVA, and UHD Forum. He is the author of numerous patents and more than 100 conference papers and journal scientific articles. He previously worked for the research Institute of Electronics and Telecommunications of Rennes (IETR) where he was a researcher in rapid prototyping of video coding standards, and he was project leader of several French and European projects. He was also a member of the research institute IRT B-COM (http://b-com.org). His interests include dataflow programming, signal processing systems and video coding. Currently, his focus is directed towards ATSC 3.0, next-generation video codecs and artificial intelligence. He served as a member of the technical committee of the Design and Implementation of Signal Processing Systems (DISPS) of the IEEE Signal Processing Society and as member a "Circuits and Systems for Video Technology" editorial board. In 2006, he received a PhD from INSA in electronic and signal processing, in collaboration with Mitsubishi Electric ITE, Rennes, France.

Received in 2022-06-22 | Approved in 2022-07-08