INVITED PAPER

Advances in Video Compression System Using Deep Neural Network: A Review and Case Studies

This article targets video coding improvements for both individual blocks of the hybrid video encoder and jointly across multiple blocks including end-to-end approaches. It reviews various modules in video coding that could benefit from neural networks and provides case studies for each of these modules.

By Dandan Ding, *Member IEEE*, Zhan Ma, *Senior Member IEEE*, Di Chen, *Member IEEE*, Qingshuang Chen, *Member IEEE*, Zoe Liu, and Fengqing Zhu, *Senior Member IEEE*

ABSTRACT | Significant advances in video compression systems have been made in the past several decades to satisfy the near-exponential growth of Internet-scale video traffic. From the application perspective, we have identified three major functional blocks, including preprocessing, coding, and postprocessing, which have been continuously investigated to maximize the end-user quality of experience (QoE) under a limited bit rate budget. Recently, artificial intelligence (AI)-powered techniques have shown great potential to further increase the efficiency of the aforementioned functional blocks, both individually and jointly. In this article, we review recent technical advances in video compression systems extensively, with an emphasis on deep neural network (DNN)-based approaches, and then present three comprehensive case studies. On preprocessing, we show a switchable texture-based video coding example that leverages DNN-based scene understanding to extract semantic areas for the improvement of a subsequent video coder. On coding, we present an end-to-end neural video coding framework that takes advantage of the stacked DNNs to efficiently and compactly code input raw videos via fully data-driven learning. On postprocessing, we demonstrate two neural adaptive filters to, respectively, facilitate the in-loop and postfiltering for the enhancement of compressed frames. Finally, a companion website hosting the contents developed in this work can be accessed publicly at https://purdueviper.github.io/dnn-coding/.

KEYWORDS | Adaptive filters; deep neural networks (DNNs); neural video coding; texture analysis.

NOMENCLATURE

AE	Autoencoder.
CNN	Convolutional neural network.
CONV	Convolution.
ConvLSTM	Convolutional LSTM.
DNN	Deep neural network.
FCN	Fully connected network.
GAN	Generative adversarial network.
LSTM	Long short-term memory.
RNN	Recurrent neural network.
VAE	Variational autoencoder.
BD-PSNR	Bjøntegaard delta PSNR.
BD-Rate	Bjøntegaard delta rate.
GOP	Group of pictures.

Manuscript received September 10, 2020; revised December 13, 2020; accepted February 8, 2021. Date of publication March 4, 2021; date of current version August 20, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 62022038 and Grant U20A20184; in part by the Zhejiang Provincial Natural Science Foundation of China under Grant LY20F010013; in part by a Google Faculty Research Award; and in part by the Google Chrome University Research Program. *(Dandan Ding and Zhan Ma contributed equally to this work.) (Corresponding author: Fengqing Zhu.)*

Dandan Ding is with the School of Information Science and Engineering, Hangzhou Normal University, Hangzhou 311121, China.

Zhan Ma is with the School of Electronic Science and Engineering, Nanjing University, Nanjing 210093, China.

Di Chen is with Google Inc., Mountain View, CA 94043 USA.

Qingshuang Chen and **Fengqing Zhu** are with the School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 USA.

Zoe Liu is with Visionular Inc., Los Altos, CA 94022 USA.

Digital Object Identifier 10.1109/JPROC.2021.3059994

MS-SSIM	Multiscale SSIM.
MSE	Mean squared error.
PSNR	Peak signal-to-noise ratio.
QP	Quantization parameter.
QoE	Quality of experience.
SSIM	Structural similarity index.
UEQ	Unequal quality.
VMAF	Video multimethod assessment fusion.
AV1	AOMedia video 1.
AVS	Audio–video standard.
H.264/AVC	H.264/advanced video coding.
H.265/HEVC	H.265/high-efficiency video coding.
VVC	Versatile video coding.
AOM	Alliance of open media.
MPEG	Moving Picture Experts Group.

I. INTRODUCTION

In recent years, Internet traffic has been dominated by a wide range of applications involving video, including video on demand (VOD), live streaming, and ultralow latency real-time communications. With ever-increasing demands in resolution (e.g., 4k, 8k, gigapixel [1], and high speed [2]) and fidelity (e.g., high dynamic range [3] and higher bit precision or bit depth [4]), more efficient video compression is imperative for content transmission and storage, by which networked video services can be successfully deployed. Fundamentally, video compression systems devise appropriate algorithms to minimize the end-to-end reconstruction distortion (or maximize the QoE), under a given bit rate budget. This is a classical rate-distortion (R-D) optimization problem. In the past, the majority of effort had been focused on the development and standardization of video coding tools for optimized R-D performance, such as the intraprediction/interprediction, transform, and entropy coding, resulting in a number of popular standards and recommendation specifications (e.g., ISO/IEC MPEG series [5]–[11], ITU-T H.26x series [9]–[13], AVS series [14]–[16], and the AV1 [17], [18] from the AOM [19]). All these standards have been widely deployed in the market and enabled advanced and high-performing services to both enterprises and consumers. They have been adopted to cover all major video scenarios from VOD, to live streaming, to ultralow latency interactive real-time communications, used for applications, such as telemedicine, distance learning, video conferencing, broadcasting, e-commerce, online gaming, and short-video platforms. Meanwhile, the system R-D efficiency can also be improved from preprocessing and postprocessing, individually and jointly, for content-adaptive encoding (CAE). Notable examples include saliency detection for subsequent regionwise quantization control and adaptive filters to alleviate compression distortions [20]–[22].

In this article, we, therefore, consider *preprocessing*, *coding*, and *postprocessing* as three basic functional blocks of an end-to-end video compression system and optimize them to provide compact and high-quality representation of input original video.

1) The "coding" block is the core unit that converts raw pixels or pixel blocks into binary bits presentation. In the past decades, the "coding" R-D efficiency has been gradually improved by introducing more advanced tools to better exploit spatial, temporal, and statistical redundancies [23]. Nevertheless, this process inevitably incurs compression artifacts, such as blockiness and ringing, due to the R-D tradeoff, especially at low bit rates.
2) The "postprocessing" block is introduced to alleviate visually perceptible impairments produced as byproducts of coding. Postprocessing mostly relies on the designated adaptive filters to enhance the reconstructed video quality or QoE. Such "postprocessing" filters can also be embedded into the "coding" loop to jointly improve reconstruction quality and R-D efficiency, for example, in-loop deblocking [24] and sample adaptive offset (SAO) [25].
3) The "preprocessing" block exploits the discriminative content preference of the human visual system (HVS), caused by the nonlinear response and frequency selectivity (e.g., masking) of visual neurons in the visual pathway. Preprocessing can extract content semantics (e.g., saliency and object instance) to improve the psychovisual performance of the "coding" block, for example, by allocating unequal qualities (UEQs) across different areas according to preprocessed cues [26].[1]

Building upon the advancements in DNNs, numerous recently created video-processing algorithms have been greatly improved to achieve superior performance, mostly leveraging the powerful nonlinear representation capacity of DNNs. At the same time, we have also witnessed an explosive growth in the invention of DNN-based techniques for video compression from both academic research and industrial practices. For example, DNN-based filtering in postprocessing was extensively studied when developing the VVC standard under the joint task force of ISO/IEC and ITU-T experts in the past three years. More recently, the standard committee issued a Call-for-Evidence (CfE) [27], [28] to encourage the exploration of deep learning-based video coding solutions beyond VVC.

In this article, we discuss recent advances in *preprocessing*, *coding*, and *postprocessing*, with a particular emphasis on the use of DNN-based approaches for efficient video compression. We aim to provide a comprehensive overview to bring readers up to date on recent advances in this emerging field. We also suggest promising directions for further exploration. As summarized in Fig. 1, we first dive into video preprocessing, emphasizing the analysis and application of content semantics, for example, saliency,

[1]Although adaptive filters can also be used in preprocessing for prefiltering, for example, denoising, motion deblurring, contrast enhancement, and edge detection, our primary focus in this work will be on semantic content understanding for subsequent intelligent "coding."

object, and texture characteristics, to video encoding. We then discuss recently developed DNN-based video coding techniques for both modularized coding tool development and end-to-end fully learned framework exploration. Finally, we provide an overview of the adaptive filters that can be either embedded in a codec loop or placed as a postenhancement to improve final reconstruction. We also present three case studies: 1) *switchable texture-based video coding* in preprocessing; 2) *E2E-NVC*; and 3) *efficient neural filtering*, to provide examples of the potential of DNNs to improve both subjective and objective efficiency over traditional video compression methodologies.

The remainder of the article is organized as follows. From Sections II–IV, we extensively review the advances in preprocessing, coding, and postprocessing, respectively. Traditional methodologies are first briefly summarized, and then DNN-based approaches are discussed in detail. As in the case studies, we propose three neural approaches in Sections V–VII, respectively. Regarding preprocessing, we develop a CNN-based texture analysis/synthesis scheme for the AV1 codec. For video compression, an end-to-end neural coding framework is developed. In our discussion of postprocessing, we present different neural methods for in-loop and postfiltering that can enhance the quality of reconstructed frames. Section VIII summarizes this work and discusses open challenges and future research directions. For your convenience, the nomenclature provides an overview of abbreviations and acronyms that are frequently used throughout this article.

II. OVERVIEW OF DNN-BASED VIDEO PREPROCESSING

Preprocessing techniques are generally applied prior to the video coding block, with the objective of guiding the video encoder to remove psychovisual redundancy and to maintain or improve visual quality, while simultaneously lowering bit rate consumption. One category of preprocessing techniques is the execution of prefiltering operations. Recently, a number of deep learning-based prefiltering approaches have been adopted for targeted coding optimization. These include denoising [29], [30], motion deblurring [31], [32], contrast enhancement [33], edge detection [34], [35], and so on. Another important topic is closely related to the analysis of video content semantics, for example, object instance, saliency attention, and texture distribution, and its application to intelligent video coding. For the sake of simplicity, we refer to this group of techniques as "preprocessing" for the remainder of this article. In our discussion below, we also limit our focus to saliency- and analysis-/synthesis-based approaches.

A. Saliency-Based Video Preprocessing

1) Saliency Prediction: Saliency is the quality of being particularly noticeable or important. Thus, the *salient area* refers to regions of an image that predominantly attracts the attention of subjects. This concept corresponds closely to the highly discriminative and selective behavior displayed in visual neuronal processing [36], [37]. Content feature extraction, activation, suppression, and aggregation also occur in the visual pathway [38].

Earlier attempts to predict saliency typically utilized handcrafted image features, such as color, intensity, and orientation contrast [39], motion contrast [40], and camera motion [41]. Later on, DNN-based semantic-level features were extensively investigated for both image content [42]–[48] and video sequences [49]–[55]. Among these features, image saliency prediction only exploits spatial information, while video saliency prediction often relies on spatial and temporal attributes jointly. One typical example of video saliency is a moving object that incurs spatiotemporal dynamics over time and is, therefore, more likely to attract users' attention. For example, Bazzani *et al.* [49] modeled the spatial relations in videos using 3-D convolutional features and the temporal consistency with a convolutional LSTM network. Bak *et al.* [50] applied a two-stream network that exploited different fusion mechanisms to effectively integrate spatial and temporal information. Sun *et al.* [51] proposed a step-gained FCN to combine the time-domain memory information and space-domain motion components. Jiang *et al.* [52] developed an object-to-motion CNN that was applied together with an LSTM network. All of these efforts to predict video saliency leveraged spatiotemporal attributes. More details regarding the spatiotemporal saliency models for video content can be found in [56].

2) Salient Object: One special example of image saliency involved the *object instance* in a visual scene, specifically, the moving object in videos. A simple, yet effective solution to the problem of predicting image saliency, in this case, involved segmenting foreground objects and background components. The segmentation of foreground objects and background components has mainly relied on foreground extraction or background subtraction. For example, motion information has been frequently used to mask out foreground objects [57]–[61].

Recently, both CNN and foreground attentive neural network (FANN) models have been developed to perform foreground segmentation [62], [63]. In addition to conventional Gaussian mixture model (GMM)-based background subtraction, recent explorations have also shown that CNN models could be effectively used for the same purpose [64], [65]. To address these separated foreground objects and background attributes, Zhang *et al.* [66] introduced a new background mode to more compactly represent background information with better R-D efficiency. To the best of our knowledge, such foreground object/background segmentation has been mostly applied in video surveillance applications, where the visual scene lends itself to easier separation.

3) Video Compression With UEQ Scales: Saliency or object, which refers to more visually attentive areas, is straightforward to apply UEQ setting in a video encoder, where light compression is used to encode the saliency

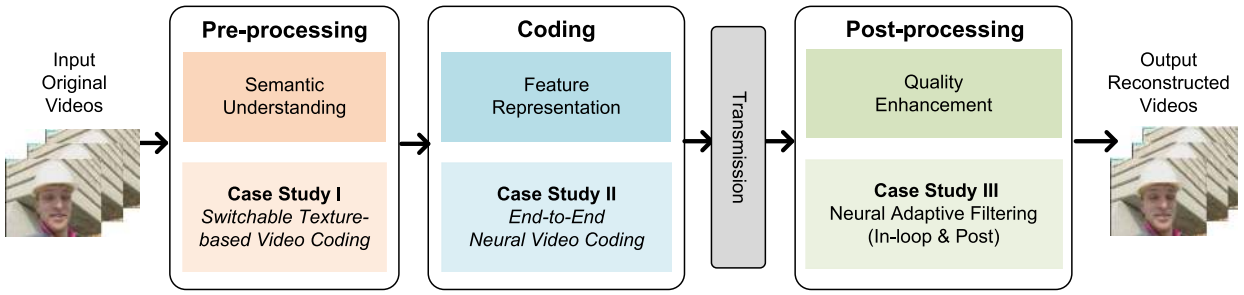

Fig. 1. ***Topic outline. This article reviews DNN-based techniques used in preprocessing, coding, and postprocessing of a practical video compression system. The "preprocessing" module leverages content semantics (e.g., texture) to guide video coding, followed by the "coding" step to represent the video content using more compact spatiotemporal features. Finally, quality enhancement is applied in "postprocessing" to improve the quality of reconstruction by alleviating processing artifacts. Companion case studies are, respectively, offered to showcase the potential of DNN algorithms in video compression.***

area, while heavy compression is used elsewhere. The use of this technique often results in a lower level of total bit rate consumption without compromising QoE.

For example, Hadi and Bajić [67] extended the well-known Itti–Koch–Niebur (IKN) model to estimate saliency in the DCT domain, also considering camera motion. In addition, saliency-driven distortion was also introduced to accurately capture the salient characteristics, in order to improve R-D optimization in H.265/HEVC. Li *et al.* [68] suggested using graph-based visual saliency to adapt the quantizations in H.265/HEVC, to reduce total bits consumption. Similarly, Ku *et al.* [69] applied saliency-weighted coding tree unit (CTU)-level bit allocation, where the CTU-aligned saliency weights were determined via low-level feature fusion.

The aforementioned methodologies rely on traditional handcrafted saliency prediction algorithms. As DNN-based saliency algorithms have demonstrated superior performance, we can safely assume that their application to video coding will lead to better compression efficiency. For example, Zhu and Xu [70] adopted a spatiotemporal saliency model to accurately control the QP in an encoder where the spatial saliency was generated using a ten-layer CNN and whose temporal saliency was calculated assuming the 2-D motion model [resulting in an average of 0.24 BD-PSNR gains over H.265/HEVC reference model (version HM16.8)]. A performance improvement due to fine-grained quantization adaptation was reported using an open-source x264 encoder in [71]. This was accomplished by jointly examining the input video frame and associated saliency maps. These saliency maps were generated by utilizing three CNN models suggested in [52], [56], and [72]. Up to 25% bit rate reduction was reported when distortion was measured using the edge-weighted SSIM. Similarly, Sun *et al.* [73] implemented a saliency-driven CTU-level adaptive bit rate control, where the static saliency map of each frame was extracted using a DNN model, and the dynamic saliency region was tracked using a moving object segmentation algorithm. Experiment results revealed that the PSNR of salient regions was improved by 1.85 dB on average.

Though saliency-based preprocessing is mainly driven by psychovisual studies, it heavily relies on saliency detection to perform UEQ-based adaptive quantization with a lower rate of bit consumption but visually identical reconstruction. On the other hand, visual selectivity behavior is closely associated with video content distribution (e.g., frequency response), leading to perceptually unequal preference. Thus, it is highly expected that such content semantics-induced discriminative features can be utilized to improve the system efficiency when integrated into the video encoder. To this end, we will discuss the analysis-/synthesis-based approach for preprocessing in Section II-B.

B. Analysis-/Synthesis-Based Preprocessing

Since most videos are consumed by human vision, subjective perception of HVS is the *best* way to evaluate quality. However, it is quite difficult to devise a profoundly accurate mathematical HVS model in an actual video encoder for rate and perceptual quality optimization, due to the complicated and unclear information processing that occurs in the human visual pathway. Instead, many pioneering psychovisual studies have suggested that neuronal response to compound stimuli is highly nonlinear [74]–[81] within the receptive field. This leads to well-known visual behaviors, such as frequency selectivity and masking, where such stimuli are closely related to the content texture characteristics. Intuitively, video scenes can be broken down into areas that are either "perceptually significant" (pSIG) (e.g., measured in an MSE sense) or "perceptually insignificant." For "perceptually insignificant" regions, users will not perceive compression or processing impairments without a side-by-side comparison with the original sample. This is because the HVS gains semantic understanding by viewing content as a whole, instead of interpreting texture details pixel by pixel [82]. This notable effect of the HVS is also referred to as "masking," where visually insignificant information, for example, perceptually insignificant pixels, will be noticeably suppressed.

In practice, we can first analyze the texture characteristics of original video content in the preprocessing step,

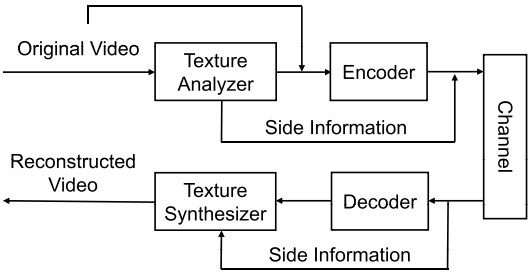

Fig. 2. ***Texture coding system. A general framework of analysis-/synthesis-based video coding.***

for example, *Texture Analyzer* in Fig. 2, in order to sort textures by their significance. Subsequently, we can use any standard compliant video encoder to encode the pSIG areas as the main bitstream payload and apply a statistical model to represent the perceptually insignificant textures with model parameters encapsulated as side information. Finally, we can use decoded areas and parsed textures to jointly synthesize the reconstructed sequences in *Texture Synthesizer*. This type of texture modeling makes good use of statistical and psychovisual representation jointly, generally requiring fewer bits, despite yielding visually identical sensation, compared to the traditional hybrid "prediction+residual" method.[2] Therefore, texture analysis and synthesis play a vital role in subsequent video coding. We will discuss related techniques in the following.

1) Texture Analysis: Early developments in texture analysis and representation can be categorized into *filter-* or *statistical modeling-based* approaches. The Gabor filter is one typical example of filter-based approaches, by which the input image is convoluted with nonlinear activation for the derivation of corresponding texture representation [84], [85]. At the same time, in order to identify static and dynamic textures for video content, Thakur and Chubach [86] utilized the 2-D dual-tree complex wavelet transform and steerable pyramid transform [87], respectively. To accurately capture the temporal variations in the video, Bansal *et al.* [88] again suggested the use of optic flow for dynamic texture indication and later synthesis, where optical flow could be generated using temporal filtering. Leveraging statistical models, such as the Markovian random field (MRF) [89], [90], is an alternative way to analyze and represent texture. For efficient texture description, such statistical modeling was then extended using handcrafted local features, for example, the scale-invariant feature transform (SIFT) [91], speeded up robust features (SURFs) [92], and local binary patterns (LBPs) [93].

Recently, stacked DNNs have demonstrated their superior efficiency in many computer vision (CV) tasks. This efficiency is mainly due to the powerful capacity of DNN features to be used for video content representation. The most straightforward scheme directly extracted features from the FC6 or FC7 layer of AlexNet [94] for texture representation. Furthermore, Cimpoi *et al.* [95] demonstrated that Fisher vectorized [96] CNN features were a decent texture descriptor candidate.

2) Texture Synthesis: Texture synthesis reverse-engineers the analysis in preprocessing to restore pixels accordingly. It generally includes both nonparametric and parametric methods. For nonparametric synthesis, texture patches are usually resampled from reference images [97]–[99]. In contrast, the parametric method utilizes statistical models to reconstruct the texture regions by jointly optimizing the observation outcomes and the model itself [87], [100], [101].

DNN-based solutions exhibit great potential for texture synthesis applications. One notable example demonstrating this potential used a pretrained image classification-based CNN model to generate texture patches [102]. Li and Wand [103] then demonstrated that a Markovian GAN-based texture synthesis could offer remarkable quality improvement.

To briefly summarize, earlier "texture analysis/synthesis" approaches often relied on handcrafted models and corresponding parameters. While they have shown good performance to some extent for a set of test videos, it is usually very difficult to generalize them to large-scale video data sets without fine-tuning parameters further. On the other hand, related neuroscience studies propose a broader definition of texture, which is more closely related to perceptual sensation, although existing mathematical or data-driven texture representations attempt to fully fulfill such perceptual motives. Furthermore, recent DNN-based schemes present a promising perspective. However, the complexity of these schemes has not yet been appropriately exploited. Thus, in Section V, we will reveal a CNN-based pixel-level texture analysis approach to segment perceptually insignificant texture areas in a frame for compression and later synthesis. In order to model the textures both spatially and temporally, we introduce a new coding mode called the "switchable texture mode" that is determined at GoP level according to the bit rate saving.

III. OVERVIEW OF DNN-BASED VIDEO CODING

A number of investigations have shown that DNNs can be used for efficient image/video coding [104]–[107]. This topic has attracted extensive attention in recent years, demonstrating its potential to enhance the conventional system with better R-D performance.

There are three major directions currently under investigation. One is resolution resampling-based video coding, by which the input videos are first downsampled prior to being encoded, and the reconstructed videos are upsampled or super-resolved to the same resolution as the input [108]–[111]. This category generally develops upscaling or super-resolution algorithms on top of

[2]A comprehensive survey of texture analysis-/synthesis-based video coding technologies can be found in [83].

standard video codecs. The second direction under investigation is modularized neural video coding (MOD-NVC), which has attempted to improve individual coding tools in a traditional hybrid coding framework using learning-based solutions. The third direction is end-to-end neural video coding (E2E-NVC), which fully leverages the stacked neural networks to compactly represent input image/video in an end-to-end learning manner. In the following, we will primarily review the latter two cases since the first one has been extensively discussed in many other studies [112].

A. Modularized Neural Video Coding

The MOD-NVC has inherited the traditional hybrid coding framework within which handcrafted tools are refined or replaced using learned solutions. The general assumption is that existing rule-based coding tools can be further improved via a data-driven approach that leverages powerful DNNs to learn robust and efficient mapping functions for more compact content representation. Two great articles have comprehensively reviewed relevant studies in this direction [106], [107]. We briefly introduce key techniques in intraprediction/interprediction, quantization, and entropy coding. Though in-loop filtering is another important piece in the "coding" block, due to its similarities with postfiltering, we have chosen to review it in quality enhancement-aimed "postprocessing" for the sake of creating a more cohesive presentation.

1) Intraprediction: Video frame content presents highly correlated distribution across neighboring samples spatially. Thus, block redundancy can be effectively exploited using causal neighbors. In the meantime, due to the presence of local structural dynamics, block pixels can be better represented from a variety of angular directed predictions.

In conventional standards, such as the H.264/AVC, H.265/HEVC, or even emerging VVC, specific prediction rules are carefully designated to use weighted neighbors for respective angular directions. From the H.264/AVC to recent VVC, intracoding efficiency has been gradually improved by allowing more fine-grained angular directions and flexible block size/partitions. In practice, an optimal coding mode is often determined by R-D optimization.

One would intuitively expect that coding performance can be further improved if better predictions can be produced. Therefore, there have been a number of attempts to leverage the powerful capacity of stacked DNNs for better intrapredictor generation, including the CNN-based predictor refinement suggested in [113] to reduce prediction residual, additional learned mode trained using FCN models reported in [114] and [115], using RNNs in [116], using CNNs in [108], even using GANs in [117], and so on. These approaches have actively utilized the neighbor pixels or blocks and/or other context information (e.g., mode) if applicable, in order to accurately represent the local structures for better prediction. Many of these approaches have reported more than 3% BD-Rate gains against the popular H.265/HEVC reference model. These examples demonstrate the efficiency of DNNs in intraprediction.

2) Interprediction: In addition to the spatial intraprediction, temporal correlations have also been exploited via interprediction, by which previously reconstructed frames are utilized to generate interpredictor for compensation using displaced motion vectors.

Temporal prediction can be enhanced using references with higher fidelity and more fine-grained motion compensation. For example, fractional-pel interpolation is usually deployed to improve prediction accuracy [118]. On the other hand, motion compensation with flexible block partitions is another major contributor to intercoding efficiency.

Similarly, earlier attempts have been made to utilize DNN solutions for better intercoding. For instance, CNN-based interpolations were studied in [119]–[121] to improve the half-pel samples. Besides, an additional virtual reference could be generated using CNN models for improved R-D decision in [122]. Xia *et al.* [123] further extended this approach using multiscale CNNs to create an additional reference closer to the current frame by which accurate pixel-wise motion representation could be used. Furthermore, conventional references could be also enhanced using DNNs to refine the compensation [124].

3) Quantization and Entropy Coding: Quantization and entropy coding are used to remove statistical redundancy. Scalar quantization is typically implemented in video encoders to remove insensitive high-frequency components, without losing the perceptual quality, while saving the bit rate. Recently, a three-layer DNN was developed to predict the local visibility threshold C_T for each CTU, by which more accurate quantization could be achieved via the connection between C_T and actual quantization step size. This development led to noticeable R-D improvement, for example, up to 11%, as reported in [125].

Context-adaptive binary arithmetic coding (CABAC) and its variants are techniques that are widely adopted to encode binarized symbols. The efficiency of CABAC is heavily reliant on the accuracy of probability estimation in different contexts. Since the H.264/AVC, handcrafted probability transfer functions (developed through exhaustive simulations and typically implemented using lookup tables) were utilized. Pfaff *et al.* [115] and Song *et al.* [126] demonstrated that a combined FCN and CNN model could be used to predict intramode probability for better entropy coding. Another example of a combined FCN and CNN model was presented in [127] to accurately encode transform indexes via stacked CNNs. Likewise, in [128], the intra-dc coefficient probability could also be estimated using DNNs for better performance.

All of these explorations have reported positive R-D gains when incorporating DNNs in traditional hybrid coding frameworks. A companion H.265-/HEVC-based software model is also offered by Liu *et al.* [106] to advance the potential for society to further pursue this line of exploration. However, integrating DNN-based tools

could exponentially increase both the computational and space complexity. Therefore, creating harmony between learning-based and conventional rule-based tools under the same framework requires further investigation. It is also worth noting that an alternative approach is currently being explored in parallel. In this approach, researchers suggest using an E2E-NVC framework to drive the raw video content representation via layered feature extraction, activation, suppression, and aggregation, mostly in a supervised learning fashion, instead of refining individual coding tools.

B. End-to-End Neural Video Coding

Representing raw video pixels as compactly as possible by massively exploiting its spatiotemporal and statistical correlations is the fundamental problem of lossy video coding. Over decades, traditional hybrid coding frameworks have utilized pixel-domain intraprediction/interprediction, transform, entropy coding, and so on to fulfill this purpose. Each coding tool is extensively examined under a specific codec structure to carefully justify the tradeoff between R-D efficiency and complexity. This process led to the creation of well-known international or industry standards, such as the H.264/AVC, H.265/HEVC, and AV1.

On the other hand, DNNs have demonstrated a powerful capacity for video spatiotemporal feature representation for vision tasks, such as object segmentation and tracking. This naturally raises the question of whether it is possible to encode those spatiotemporal features in a compact format for efficient lossy compression.

Recently, we have witnessed the growth of video coding technologies that rely completely on end-to-end supervised learning. Most learned schemes still closely follow the conventional intra/interframe definition by which different algorithms are investigated to efficiently represent the intraspatial textures, intermotion, and the interresiduals (if applicable) [104], [129]–[131]. Raw video frames are fed into stacked DNNs to extract, activate, and aggregate appropriate compact features (at the bottleneck layer) for quantization and entropy coding. Similarly, R-D optimization is also facilitated to balance the rate and distortion tradeoff. In the following, we will briefly review the aforementioned key components.

1) Nonlinear Transform and Quantization: The AE or VAE architectures are typically used to transform the intratexture or interresidual into compressible features. For example, Toderic *et al.* [132] first applied fully connected *recurrent AEs* for variable-rate thumbnail image compression. Their work was then improved in [133] and [134] with the support of full-resolution image, unequal bit allocation, and so on. Variable bit rate is intrinsically enabled by these recurrent structures. The recurrent AEs, however, suffer from higher computational complexity at higher bit rates because more recurrent processing is desired. Alternatively, *convolutional AEs* have been extensively studied in the past years, where different bit rates are adapted by setting a variety of λ's to optimize the R-D tradeoff. Note that different network models may be required for individual bit rates, making hardware implementation challenging (e.g., model switch from one-bit rate to another). Recently, conditional convolution [135] and scaling factor [136] were proposed to enable variable-rate compression using a single or very limited network model without noticeable coding efficiency loss, which makes the convolutional AEs more attractive for practical applications.

To generate a more compact feature representation, Balle *et al.* [105] suggested replacing the traditional nonlinear activation, for example, ReLU, using generalized divisive normalization (GDN) that is theoretically proved to be more consistent with human visual perception. A subsequent study [137] revealed that GDN outperformed other nonlinear rectifiers, such as ReLU, leakyReLU, and tan h, in compression tasks. Several follow-up studies [138], [139] directly applied GDN in their networks for compression exploration.

Quantization is a nondifferentiable operation, basically converting arbitrary elements into symbols with a limited alphabet for efficient entropy coding in compression. Quantization must be derivable in the end-to-end learning framework for backpropagation. A number of methods, such as adding uniform noise [105], stochastic rounding [132], and soft-to-hard vector quantization [140], were developed to approximate a continuous distribution for differentiation.

2) Motion Representation: Chen *et al.* [104] developed the DeepCoder where a simple convolutional AE was applied for both intracoding and residual coding at fixed 32×32 blocks, and block-based motion estimation in traditional video coding was reused for temporal compensation. Lu *et al.* [141] introduced the optical flow for motion representation in their DVC work, which, together with the intracoding in [142], demonstrated similar performance compared with the H.265/HEVC. However, coding efficiency suffered from a sharp loss at low bit rates. Chen *et al.* [136] extended their nonlocal attention optimized image compression (NLAIC) for intraencoding and residual encoding and applied second-order flow-to-flow prediction for more compact motion representation, showing consistent R-D gains across different contents and bit rates.

Motion can also be implicitly inferred via temporal interpolation. For example, Wu *et al.* [143] applied RNN-based frame interpolation. Together with the residual compensation, RNN-based frame interpolation offered comparable performance to the H.264/AVC. Djelouah *et al.* [144] furthered interpolation-based video coding by utilizing advanced optical flow estimation and feature domain residual coding. However, temporal interpolation usually led to an inevitable structural coding delay.

Another interesting exploration made by Ripple *et al.* in [130] was to jointly encode motion flow and residual

using compound features, where a recurrent state was embedded to aggregate multiframe information for efficient flow generation and residual coding.

3) R-D Optimization: Li *et al.* [145] utilized a separate three-layer CNN to generate an importance map for spatial-complexity-based adaptive bit allocation, leading to noticeable subjective quality improvement. Mentzer *et al.* [140] further utilized the masked bottleneck layer to unequally weight features at different spatial locations. Such importance map embedding is a straightforward approach to end-to-end training. Importance derivation was later improved with the nonlocal attention [146] mechanism to efficiently and implicitly capture both global and local significance for better compression performance [136].

Probabilistic models play a vital role in data compression. Assuming the Gaussian distribution for feature elements, Ballé *et al.* [142] utilized hyperpriors to estimate the parameters of the Gaussian scale model (GSM) for latent features. Later, Hu *et al.* [147] used hierarchical hyperpriors (coarse-to-fine) to improve the entropy models in multiscale representations. Minnen *et al.* [148] improved the context modeling using joint autoregressive spatial neighbors and hyperpriors based on the GMM. Autoregressive spatial priors were commonly fused by PixelCNNs or PixelRNNs [149]. Reed *et al.* [150] further introduced multiscale PixelCNNs, yielding competitive density estimation and great boost in speed [e.g., from $O(N)$ to $O(\log N)$]. Prior aggregation (PA) was later extended from 2-D architectures to 3-D PixelCNNs [140]. Channelwise weights sharing-based 3-D implementations could greatly reduce network parameters without performance loss. Parallel 3-D PixelCNNs for practical decoding were presented by Chen *et al.* [136]. Previous methods accumulated all the priors to estimate the probability based on a single GMM assumption for each element. Recent studies in [151] and [152] have shown that weighted GMMs can further improve coding efficiency.

Pixel error, such as MSE, was one of the most popular loss functions used. Concurrently, SSIM (or MS-SSIM) was also adopted because of its greater consistency with visual perception. Simulations revealed that SSIM-based loss can improve reconstruction quality, especially at low bit rates. Toward the perceptual-optimized encoding, perceptual losses that were measured by adversarial loss [153]–[155] and VGG loss [156] were embedded in learning to produce visually appealing results.

Though E2E-NVC is still in its infancy, its fast-growing R-D efficiency holds a great deal of promise. This is especially true, given that we can expect neural processors to be deployed massively in the near future [157].

IV. OVERVIEW OF DNN-BASED POSTPROCESSING

Compression artifacts are inevitably present in both traditional hybrid coding frameworks and learned compression approaches, for example, blockiness, ringing, and cartoonishness, severely impairing visual sensation and QoE. Thus, quality enhancement filters are often applied as a postfiltering step or in-loop module to alleviate compression distortions. Toward this goal, adaptive filters are usually developed to minimize the error between original and distorted samples.

A. In-Loop Filtering

Existing video standards are mainly utilizing the in-loop filters to improve the subjective quality of reconstruction and also to offer better R-D efficiency due to enhanced references. Examples include deblocking [24], SAO [25], constrained directional enhancement filter (CDEF) [158], loop-restoration (LR) [159], adaptive loop filter (ALF) [160], and so on.

Recently, numerous CNN models have been developed for in-loop filtering via a data-driven approach to learn the mapping functions. It is worth pointing out that prediction relationships must be carefully examined when designing in-loop filters due to the frame referencing structure and potential error propagation. Earlier explorations of this subject have mainly focused on designing DNN-based filters for intracoded frames, particularly by trading network depth and parameters for better coding efficiency. For example, IFCNN [161] and VRCNN [162] are shallow networks with ≈50 000 parameters, providing up to 5% BD-Rate savings for the H.265/HEVC intraencoder. More gains can be obtained if we use a deeper and denser network [163]–[165], for example, 5.7% BD-Rate gain reported in [163] by using the model with 3 340 000 parameters and 8.50% BD-Rate saving obtained in [166] by using the model with 2 298 160 parameters. The more parameters a model has, the more complex it is. Unfortunately, greater complexity limits the network's potential for practical application. Such intraframe-based in-loop filters treat decoded frames equally, without the consideration of in-loop interprediction dependence. Nevertheless, the aforementioned networks can be used in postfiltering out of the coding loop.

It is necessary to include temporal prediction dependence while designing the in-loop CNN-based filters for interframe coding. Some studies leveraged prior knowledge from the encoding process to assist the CNN training and inference. For example, Jia *et al.* [167] incorporated the colocated block information for in-loop filtering. Meng *et al.* [168] utilized the coding unit partition for further performance improvement. Li and Yu [169] input both the reconstructed frame and the difference between the reconstructed and predicted pixels to improve the coding efficiency. Applying prior knowledge in learning may improve the coding performance, but it further complicates the CNN model by involving additional information in the networks. On the other hand, the contribution of this prior knowledge is quite limited because such additional priors are already implicitly embedded in the reconstructed frame.

If CNN-based in-loop filtering is applied to frame I_0, the impact will be gradually propagated to frame I_1 that has frame I_0 as the reference. Subsequently, I_1 is the reference of I_2 and so forth.[3] If frame I_1 is filtered again by the same CNN model, an overfiltering problem will be triggered, resulting in severely degraded performance, as analyzed in [170]. To overcome this challenging problem, a CNN model called SimNet was built to carry the relationship between the reconstructed frame and its original frame in [171] to adaptively skip filtering operations in intercoding. SimNet reported 7.27% and 5.57% BD-Rate savings for intracoding and intercoding of AV1, respectively. A similar skipping strategy was suggested by Chen *et al.* [172] to enable a wide activation residual network (WARN), yielding 14.42% and 9.64% BD-Rate savings for respective intracoding and intercoding on the AV1 platform.

Alternative solutions resort to the more expensive R-D optimization to avoid the overfiltering problem. For example, Yin *et al.* [173] developed three sets of CNN filters for luma and chroma components, where the R-D optimal CNN model is used and signaled in the bitstream. Similar ideas are developed in [174] and [175] as well, in which multiple CNN models are trained and the R-D optimal model is selected for inference.

It is impractical to use deeper and denser CNN models in applications. It is also very expensive to conduct R-D optimization to choose the optimal one from a set of pretrained models. Note that a limited number of pretrained models are theoretically insufficient to be generalized for large-scale video samples. To this end, in Section VII-A, we introduce a guided-CNN scheme that adapts shallow CNN models according to the characteristics of input video content.

B. Postfiltering

Postfiltering is generally applied to the compressed frames at the decoder side to further enhance the video quality for better QoE.

Previous in-loop filters designated for intracoded frames can be reused for *single-frame* postfiltering [162], [176]–[184]. Appropriate retraining may be applied in order to better capture the data characteristics. However, single-frame postfiltering may introduce quality fluctuation across frames. This may be due to the limited capacity of CNN models to deal with a great number of video contents. Thus, *multiframe* postfiltering can be devised to massively exploit the correlation across successive temporal frames. By doing so, it not only greatly improves the single-frame solution, but also offers better temporal quality over time.

Typically, a two-step strategy is applied for multiframe postfiltering. First, neighboring frames are aligned to the current frame via (pixel-level) motion estimation and compensation (MEMC). Then, all aligned frames are fed into networks for high-quality reconstruction. Thus, the accuracy of MEMC greatly affects reconstruction performance. In applications, learned optical flow, such as FlowNet [185], FlowNet2 [186], PWC-Net [187], and TOFlow [188], are widely used.

Some exploration has already been made in this arena: Bao *et al.* [189] and Wang *et al.* [190] implemented a general video quality enhancement framework for denoising, deblocking, and super-resolution, where Bao *et al.* [189] employed the FlowNet and Wang *et al.* [190] used pyramid, cascading, and deformable convolutions to, respectively, align frames temporally. Meanwhile, Yang *et al.* [191] proposed a multiframe quality enhancement framework called MFQE-1.0, in which a spatial transformer motion compensation (STMC) network is used for alignment, and a deep quality enhancement network (QE-net) is employed to improve reconstruction quality. Then, Guan *et al.* [192] upgraded MFQE-1.0 to MFQE-2.0 by replacing QE-net using a dense CNN model, leading to better performance and less complexity. Later on, Tong *et al.* [193] suggested using FlowNet2 for temporal frame alignment (instead of default STMC), yielding 0.23-dB PSNR gain over the original MFQE-1.0. Similarly, FlowNet2 is also used in [194] for improved efficiency.

All of these studies suggested the importance of temporal alignment in postfiltering. Thus, in the subsequent case study (see Section VII-B), we first examine the efficiency of alignment and then further discuss the contributions from respective intracoded and intercoded frames for the quality enhancement of final reconstruction. This will help audiences gain a deeper understanding of similar postfiltering techniques.

V. CASE STUDY FOR PREPROCESSING: SWITCHABLE TEXTURE-BASED VIDEO CODING

This section presents a switchable texture-based video preprocessing that leverages DNN-based semantic understanding for subsequent coding improvement. In short, we exploit DNNs to accurately segment "perceptually InSIGnifcant" (pInSIG) texture areas to produce a corresponding pInSIG mask. In many instances, this mask drives the encoder to perform separately for pInSIG textures that are typically inferred without additional residuals and "pSIG" areas elsewhere using the traditional hybrid coding method. This approach is implemented on top of the AV1 codec [195]–[197] by enabling the GoP-level switchable mechanism, resulting in noticeable bit rate savings for both standard test sequences and additional challenging sequences from the YouTube UGC data set [198], under similar perceptual quality. The method that we propose is a pioneering work that integrates learning-based texture analysis and reconstruction approaches with modern video codec to enhance video compression performance.

[3]Even though more advanced interreferencing strategies can be devised, interpropagation-based behavior remains the same.

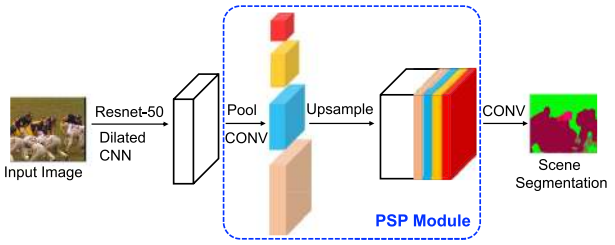

Fig. 3. ***Texture analyzer. The proposed semantic segmentation network using PSPNet [199] and ResNet-50 [200].***

A. Texture Analysis

Our previous attempt [201] yielded encouraging bit rate savings without decreasing visual quality. This was accomplished by perceptually differentiating pInSIG textures and other areas to be encoded in a hybrid coding framework. However, the corresponding texture masks were derived using traditional methods, at the coding block level. On the other hand, building upon advancements created by DNNs and large-scale labeled data sets (e.g., ImageNet [202], COCO [203], and ADE20K [204]), learning-based semantic scene segmentation algorithms [199], [204], [205] have been tremendously improved to generate accurate pixel-level texture masks.

In this work, we first rely on the powerful ResNet50 [200] with dilated convolutions [206], [207] to extract feature maps that effectively embed the content semantics. We then introduce the pyramid pooling module from PSPNet [199] to produce a pixel-level semantic segmentation map shown in Fig. 3. Our implementation starts with a pretrained PSPNet model generated using the MIT SceneParse150 [208] as a scene parsing benchmark. We then retrain the model on a subset of a densely annotated data set ADE20K [204]. In the end, the model offers a pixel segmentation accuracy of 80.23%.

It is worthwhile to note that such pixel-level segmentation may result in the creation of a number of semantic classes. Nevertheless, this study suggests grouping similar texture classes commonly found in nature scenes together into four major categories, for example, "earth and grass," "water, sea, and river," "mountain and hill," and "tree." Each texture category would have an individual segmentation mask to guide the compression performed by the succeeding video encoder.

B. Switchable Texture-Based Video Coding

Texture masks are generally used to identify texture blocks and perform the encoding of texture blocks and nontexture blocks separately, as illustrated in Fig. 4(a). In this case study, the AV1 reference software platform is selected to exemplify the efficiency of our proposal.

1) Texture Blocks: Texture and nontexture blocks are identified by overlaying the segmentation mask from the texture analyzer on its corresponding frame. These frame-aligned texture masks produce pixel-level accuracy, which is capable of supporting arbitrary texture shapes. However, in order to support the block processing commonly adopted by video encoders, we propose refining original pixel-level masks to their block-based representations. The minimum size of a texture block is 16×16. In order to avoid boundary artifacts and maintain temporal consistency, we implemented a conservative two-step strategy to determine the texture block. First, the block itself must be fully contained in the texture region marked using the pixel-level mask. Then, its warped representation to temporal references (e.g., the preceding and succeeding frames in the encoding order) has to be inside the masked texture area of corresponding reference frames as well. Finally, these texture blocks are encoded using the *texture mode*, and nontexture blocks are encoded as usual using the hybrid coding structure.

2) Texture Mode: A coded block of the texture mode is inferred by its temporal reference using the global motion parameters without incurring any motion compensation residuals. In contrast, nontexture blocks are compressed using a hybrid "prediction+residual" scheme. For each current frame and any one of its reference frames, AV1 syntax specifies only one set of global motion parameters at the frame header. Therefore, to comply with the AV1 syntax,

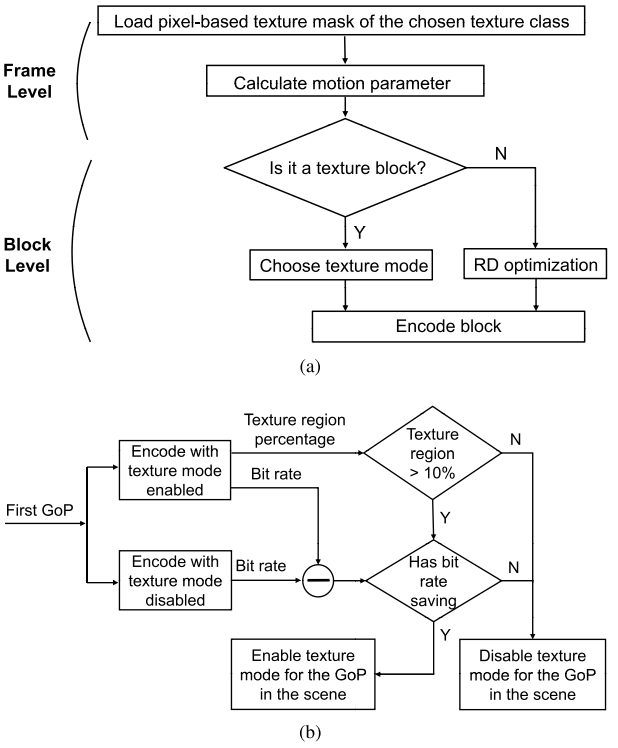

Fig. 4. ***Texture mode and switchable control scheme. (a) Texture mode encoder implementation. (b) Switchable texture mode decision.***

our implementation only considers one texture class for each frame. This guarantees the general compatibility of our solution to existing AV1 decoders. We further modify the AV1 global motion tool to estimate the motion parameters based on the texture regions of the current frame and its reference frame. We use the same feature extraction and model fitting approach as in the global motion coding tool in order to provide a more accurate motion model for the texture regions. This is done to prevent visual artifacts on the block edges between the texture and nontexture blocks in the reconstructed videos. Although we have demonstrated our algorithms using the AV1 standard, we expect that the same methodology can be applied to other standards. For instance, when using the H.265/HEVC standard, we can leverage the SKIP mode syntax to signal the texture mode instead of utilizing the global motion parameters.

Previous discussions have suggested that the texture mode is enabled along with interprediction. Our extensive studies have also demonstrated that it is better to activate the texture mode in frames where bidirectional predictions are allowed (e.g., B-frames), for the optimal tradeoff between bit rate saving and perceived quality. As will be shown in the following performance comparisons, we use an eight-frame GoP (or golden-frame (GF) group defined in AV1) to exemplify the texture mode in every other frame, by which the compound prediction from bidirectional references can be facilitated for prediction warping. Such bidirectional prediction could also alleviate possible temporal quality flickering.

3) Switchable Optimization: In our previous work [209], the texture mode was enabled for every B frame, demonstrating significant bit rate reduction at the same level of perceptual sensation in most standard test videos, in comparison to the AV1 anchor. However, some videos did cause the model to perform more poorly. One reason for this effect is that higher QP settings typically incur more all-zero residual blocks. Alternatively, the texture mode is also content-dependent: a relatively small number of texture blocks may be present for some videos. Both scenarios limit the bit rate savings, and overhead of extra bits is mandatory for global motion signaling if texture mode is enabled.

To address these problems, we introduce a switchable scheme to determine whether texture mode could be potentially enabled for a GoP or a GF group. The criteria for switching are based on the texture region percentage that is calculated as the average ratio of texture blocks in B-frames and on the potential bit rate savings with or without texture mode. Fig. 4(b) illustrates the switchable texture mode decision. Currently, we use bit rate saving as a criterion for switch decisions when the texture mode is enabled. This assumes that perceptual sensation will remain nearly the same since these texture blocks are perceptually insignificant.

C. Experimental Results

We selected sequences with texture regions from standard test sequences and the more challenging YouTube UGC data set[4] [198]. The YouTube UGC data set is a sample selected from thousands of user-generated content (UGC) videos uploaded to YouTube. The names of the UGC videos follow the format of Category_Resolution_UniqueID. We calculate the bit rate savings at different QP values for 150 frames of the test sequences. In our experiments, we use the following parameters for the AV1 codec[5] as the baseline: eight-frame GoP or GF group using random access configuration; 30 FPS; constant quality rate control policy; multilayer coding structure for all GF groups; and maximum intraframe interval at 150. We evaluate the performance of our proposed method in terms of bit rate savings and perceived quality.

1) Coding Performance: To evaluate the performance of the proposed switchable texture mode method, bit rate savings at four quantization levels (QP = 16, 24, 32, and 40) are calculated for each test sequence in comparison to the AV1 baseline.

a) Texture analysis: We compare two DNN-based texture analysis methods [209], [211] with a handcrafted feature-based approach [210] for selected standard test sequences. Results are shown in Table 1. A positive bit rate saving (%) indicates a reduction compared with the AV1 baseline. Compared to the feature-based approach, DNN-based methods show improved performance in terms of bit rate saving. The feature-based approach relies on color and edge information to generate the texture mask and is less accurate and consistent both spatially and temporally. Therefore, the number of blocks that are reconstructed using texture mode is usually much smaller than that of DNN-based methods. Note that the parameters used in the feature-based approach require manually tuning for each video to optimize the texture analysis output. The pixel-level segmentation [209] shows further advantages compared with the block-level method [211] since the CNN model does not require block size to be fixed.

b) Switchable scheme We also compare the proposed method, also known as, *tex-switch*, with our previous work in [209], also known as, *tex-allgf*, which enables texture mode for all frames in a GF group. Both methods use the same encoder setting for a fair comparison. Bit rate saving results for various videos at different resolutions against the AV1 baseline are shown in Table 2. A positive bit rate saving (%) indicates a reduction compared with the AV1 baseline.

In general, compared to the AV1 baseline, the coding performance of *tex-allgf* shows significant bit rate savings at lower QPs. However, as QP increases, the savings are diminished. In some cases, *tex-allgf* exhibits poorer coding performance than the AV1 baseline at a high QP

[4]https://media.withyoutube.com/

[5]AV1 codec change-Id: Ibed6015aa7cce12fcc6f314ffde76624df4ad2a1.

Table 1 Bit Rate Saving (%) Comparison Between Handcraft Feature (FM) [210], Block-Level DNN (BM) [211], and Pixel-Level DNN (PM) [209] Texture Analysis Against the AV1 Baseline for Selected Standard Test Sequences Using *tex-allfg* Method

Video Sequence	QP=16 (%)			QP=24 (%)			QP=32 (%)			QP=40 (%)		
	FM	BM	PM	FM	BM	PM	FM	BM	PM	FM	BM	PM
Coastguard	−0.17	7.80	9.14	−0.36	6.99	8.01	−0.43	4.70	5.72	−0.62	1.90	2.13
Flower	7.42	10.55	13.00	5.42	8.66	10.78	2.51	5.96	4.95	0.19	3.38	1.20
Waterfall	3.65	4.63	13.11	1.58	3.96	7.21	−0.14	−0.33	1.30	−3.00	−3.74	−3.48
Netflix_aerial	1.15	8.59	9.15	−0.26	2.15	5.59	−1.32	−0.68	1.05	−2.10	−4.59	−4.01
Intotree	0.88	5.32	9.71	0.15	4.32	9.42	−0.14	1.99	8.46	−0.26	−2.83	4.92

(e.g., negative numbers at QP 40). At a high QP, most blocks have zero residual due to heavy quantization, leading to very limited margins for bit rate savings using texture mode. In addition, few extra bits are required for the signaling of global motion of texture mode coded blocks. The bit savings gained through residual skipping in texture mode still cannot compensate for the bits used as overhead for the side information.

Furthermore, the proposed *tex-switch* method retains the greatest bit rate savings offered by *tex-allgf* and resolves the loss at higher QP settings. As shown in Table 2, negative numbers are mostly removed (highlighted in green) by the introduction of a GoP-level switchable texture mode. In some cases where *tex-switch* has zero bit rate savings compared to the AV1 baseline, the texture mode is completely disabled for all the GF groups, whereas *tex-allgf* has a loss. In a few cases, however, *tex-switch* has less bit rate saving than *tex-allgf* (highlighted in red). This is because the bit rate saving performance of the first GF group in the scene fails to accurately represent the whole scene in some of the UGC sequences with short scene cuts. A possible solution is to identify additional GF groups that show potential bit rate savings and enable texture mode for these GF groups.

2) Subjective Evaluation: Although significant bit rate savings have been achieved compared to the AV1 baseline, it is acknowledged that identical QP values do not necessarily imply the same video quality. We have performed a subjective visual quality study with 20 participants. Reconstructed videos produced by the proposed method (*tex-switch*) and the baseline AV1 codec at QP = 16, 24, 32, and 40 are arranged randomly and assessed by the participants using a double stimulus continuous quality scale (DSCQS) method [212]. Subjects have been asked to choose among three options: the first video has better visual quality, the second video has better visual quality, or there is no difference between the two versions.

The result of this study is summarized in Fig. 5. The "Same Quality" indicates the percentage of participants that cannot tell the difference between the reconstructed videos by the AV1 baseline codec and the proposed method *tex-switch* (69.03% on average). The term "tex-switch" indicates the percentage of participants that prefer the reconstructions by the proposed method *tex-switch* (14.32% on average); the "AV1" indicates the percentage of participants who think the visual quality of the reconstructed videos using the AV1 baseline is better (16.65% on average).

Table 2 Bit Rate Saving (%) Comparison for *tex-allgf* and *tex-switch* Methods Against the AV1 Baseline

Resolution	Video Sequence	QP=16 (%)		QP=24 (%)		QP=32 (%)		QP=40 (%)	
		tex-allgf	*tex-switch*	*tex-allgf*	*tex-switch*	*tex-allgf*	*tex-switch*	*tex-allgf*	*tex-switch*
CIF	Bridgeclose	15.78	15.78	10.87	10.87	4.21	4.21	2.77	2.77
	Bridgefar	10.68	10.68	8.56	8.56	6.34	6.34	6.01	6.01
	Coastguard	9.14	9.14	8.01	8.01	5.72	5.72	2.13	2.13
	Flower	13.00	13.00	10.78	10.78	4.95	4.95	1.20	1.20
	Waterfall	13.11	13.11	7.21	7.21	1.30	1.30	−3.48	0.00
512×270	Netflix_ariel	9.15	9.15	5.59	5.59	1.05	1.05	−4.01	0.00
360P	NewsClip_360P-1e1c	10.77	10.77	9.27	9.27	5.23	5.23	1.54	1.54
	NewsClip_360P-22ce	17.37	17.37	15.79	15.79	16.37	16.37	17.98	17.98
	TelevisionClip_360P-3b9a	1.45	1.45	0.48	0.48	−1.09	0.00	−3.26	0.00
	TelevisionClip_360P-74dd	1.66	1.66	1.17	1.17	0.36	0.36	−0.37	0.00
480P	HowTo_480P-04f1	3.81	3.81	2.57	2.57	0.93	0.93	0.06	0.36
	HowTo_480P-4c99	2.36	2.36	1.67	1.67	0.37	0.00	−1.16	0.00
	MusicVideo_480P-1eee	3.31	3.31	3.29	3.29	2.53	2.53	−0.30	−0.30
	NewsClip_480P-15fa	6.31	6.31	6.05	5.79	0.53	0.11	−0.79	0.03
	NewsClip_480P-7a0d	11.54	11.54	10.03	10.03	1.53	1.53	0.08	0.00
	TelevisionClip_480P-19d3	3.13	3.13	2.86	2.86	1.66	1.66	0.58	0.00
720P	HowTo_720P-0b01	12.72	12.72	11.84	11.84	9.31	9.31	6.35	6.35
	MusicVideo_720P-3698	1.76	1.76	1.07	1.07	0.30	0.30	−0.17	0.00
	MusicVideo_720P-4ad2	6.93	6.93	3.81	3.81	1.87	1.87	0.60	0.11
1080P	HowTo_1080P-4d7b	7.31	7.31	6.07	6.07	3.21	3.21	0.72	0.72
	MusicVideo_1080P-55af	3.88	3.88	1.78	1.78	0.31	0.33	−0.99	−0.68
	intotree	9.71	9.71	9.42	9.42	8.46	8.46	4.92	4.92
	Average	7.96	7.96	6.28	6.27	3.38	3.40	1.45	2.05

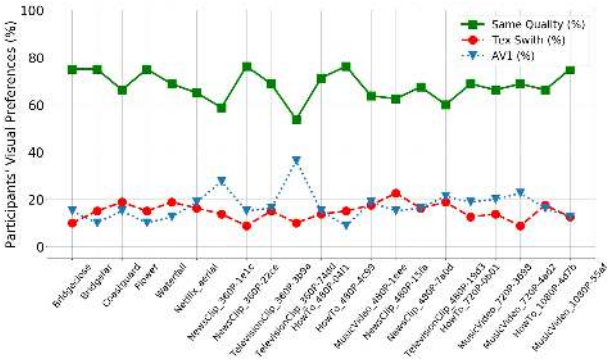

Fig. 5. ***Subjective evaluation of visual preference. Results show average subjective preference (%) for QP = 16, 24, 32, and 40 compared between AV1 baseline and proposed switchable texture mode.***

We observe that the results are sequence-dependent, and both spatial and temporal artifacts can appear in the reconstructed videos. The main artifacts come from the inaccurate pixel-based texture mask. For example, in some frames of *TelevisionClip_360P-74dd* sequence, the texture masks include parts of the moving objects in the foreground, which are reconstructed using the texture mode. Since the motion of the moving objects is different from the motion of the texture area, there are noticeable artifacts around those parts of the frame. To further improve the accuracy of region analysis using DNN-based preprocessing, we plan to incorporate an in-loop perceptual visual quality metric for optimization during the texture analysis and reconstruction.

D. Discussion and Future Direction

We proposed a DNN-based texture analysis/synthesis coding tool for the AV1 codec. Experimental results show that our proposed method can achieve noticeable bit rate reduction with satisfying visual quality for both standard test sets and UGCs, which is verified by a subjective study. We envision that video coding driven by semantic understanding will continue to improve in terms of both quality and bit rate, especially by leveraging advances of deep learning methods. However, there remain several open challenges that require further investigation.

Accuracy of the region analysis is one of the major challenges for integrating semantic understanding into video coding. However, recent advances in scene understanding have significantly improved the performance of the region analysis. Visual artifacts are still noticeable when a nontexture region is incorrectly included in the texture mask, particularly if the analysis/synthesis coding system is open loop. One potential solution is to incorporate some perceptual visual quality measures in-loop during the texture region reconstruction.

Benchmark video segmentation data sets are important for developing machine learning methods for video-based semantic understanding. Existing segmentation data sets are either based on images with texture [213], contain general video objects only [214], [215], or focus on visual quality but lack segmentation ground truth.

VI. CASE STUDY FOR CODING: END-TO-END NEURAL VIDEO CODING

This section presents a framework for E2E-NVC. We include a discussion of its key components as well as its overall efficiency. Our proposed method is extended from our pioneering work in [104] but with significant performance improvements by allowing fully end-to-end learning-based spatiotemporal feature representation. More details can be found in [131], [136], and [216].

A. Framework

As with all modern video encoders, the proposed E2E-NVC compresses the first frame in each GoP as an intraframe using a VAE-based compression engine (neuro-Intra). It codes the remaining frames in each GoP using motion-compensated prediction. As shown in Fig. 6(a), the proposed E2E-NVC uses the VAE compressor (neuro-Motion) to generate the multiscale motion field between the current frame and the reference frame. Then, a multiscale motion compensation network (MS-MCN) takes multiscale compressed flows, warps the multiscale features of the reference frame, and combines these warped features to generate the predicted frame. The prediction residual is then coded using another VAE-based compressor (neuro-Res).

A low-delay E2E-NVC-based video encoder is specifically illustrated in this work. Given a GoP $\mathbb{X} = \{\mathbf{X_1}, \mathbf{X_2}, \ldots, \mathbf{X_t}\}$, we first encode $\mathbf{X_1}$ using the neuro-Intra-module and have its reconstructed frame $\hat{\mathbf{X}}_1$. The following frame X_2 is encoded predictively, using neuro-Motion, MS-MCN, and neuro-Res together, as shown in Fig. 6(a). Note that MS-MCN takes the multiscale optical flows $\{\vec{f}_d^1, \vec{f}_d^2, \ldots, \vec{f}_d^s\}$ derived by the pyramid decoder in neuro-Motion and then uses them to generate the predicted frame $\hat{\mathbf{X}}_2^p$ by multiscale motion compensation. Displaced interresidual $\mathbf{r}_2 = \mathbf{X}_2 - \hat{\mathbf{X}}_2^p$ is then compressed in neuro-Res, yielding the reconstruction $\hat{\mathbf{r}}_2$. The final reconstruction $\hat{\mathbf{X}}_2$ is given by $\hat{\mathbf{X}}_2 = \hat{\mathbf{X}}_2^p + \hat{\mathbf{r}}_2$. All of the remaining P-frames in the GoP are then encoded using the same procedure.

Fig. 6(b) illustrates the general architecture of the VAE model. The VAE model includes the main encoder–decoder pair that is used for latent feature analysis and synthesis, as well as a hyper-encoder–decoder for hyperprior generation. The main encoder $\mathbf{E}_m$ uses four stacked CNN layers. Each convolutional layer employs stride convolutions to achieve downsampling (at a factor of 2 in this example) and cascaded convolutions for efficient feature extraction (here, we use three ResNet-based residual blocks [200]).[6]

[6]We choose to apply cascaded ResNets for stacked CNNs because they are highly efficient and reliable. Other efficient CNN architectures could also be applied.

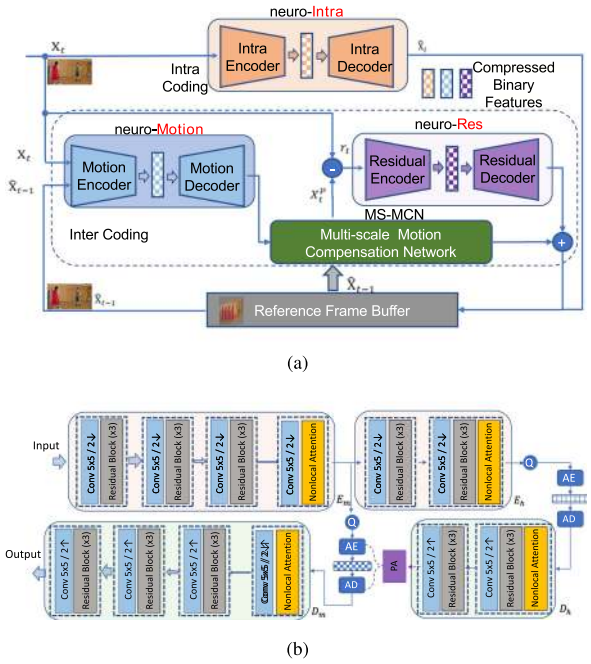

Fig. 6. ***E2E-NVC. This E2E-NVC in (a) consists of modularized intracoding and intercoding, where intercoding utilizes respective motion and residual coding. Each component is well exploited using a stacked CNN-based VAE for efficient representations of intrapixels, displaced interresiduals, and intermotions. All modularized components are interconnected and optimized in an end-to-end manner. (b) General VAE model applies stacked convolutions (e.g., 5×5) with main encoder–decoder (E_m, D_m) and hyper-encoder–decoder pairs (E_h, D_h), where the main encoder E_m includes four major convolutional layers (e.g., convolutional downsampling and three residual blocks (×3) robust feature processing [200]). Hyperdecoder D_h mirrors the steps in hyperencoder E_h for hyperprior information generation. PA engine collects the information from hyperprior, autoregressive spatial neighbors, and temporal correspondences (if applicable) for the main decoder D_m to reconstruct the input scene. Nonlocal attention is adopted to simulate the saliency masking at bottlenecks, and the rectified linear unit (ReLU) is implicitly embedded with convolutions for enabling the nonlinearity. "Q" is for quantization. AE and AD are for respective arithmetic encoding and decoding. 2↓ and 2↑ are downsampling and upsampling at a factor of 2 for both horizontal and vertical dimensions.***

We use two-layer hyperencoder $\mathbf{E}_h$ to further generate the subsequent hyperpriors as side information, which is used in the entropy coding of the latent features.

We apply stacked convolutional layers with a limited (3×3) receptive field to capture the spatial locality. These convolutional layers are stacked in order to simulate layerwise feature extraction. These same ideas are used in many relevant studies [142], [148]. We utilize the simplest ReLU as the nonlinear activation function (although other nonlinear activation functions, such as the GDN in [105], could be used as well).

The HVS operates in two stages. First, the observer scans an entire scene to gain a complete understanding of everything within the field of vision. Second, the observer focuses their attention on specific salient regions. During image and video compression, this mechanism of visual attention can be used to ensure that bit resources are allocated where they are most needed (e.g., via unequal feature quantization) [140], [217]. This allows resources to be assigned such that salient areas are more accurately reconstructed, while resources are conserved in the reconstruction of less-salient areas. To more accurately discern salient from nonsalient areas, we adopt the nonlocal attention module (NLAM) at the bottleneck layers of both the main encoder and hyperencoder, prior to quantization, in order to include both global and local information.

To enable more accurate conditional probability density modeling for entropy coding of the latent features, we introduce the *PA* engine that fuses the inputs from the hyperpriors, spatial neighbors, and temporal context (if applicable).[7] Information theory suggests that more accurate context modeling requires fewer resources (e.g., bits) to represent information [218]. For the sake of simplicity, we assume the latent features (e.g., motion, image pixel, and residual) are following the Gaussian distribution as in [147] and [148]. We use the PA engine to derive the mean and standard deviation of the distribution for each feature.

B. Neural Intracoding

Our neuro-Intra is a simplified version of the NLAIC that was originally proposed in [136].

One major difference between the NLAIC and the VAE model using autoregressive spatial context in [148] is the introduction of the NLAM inspired by Zhang *et al.* [219]. In addition, we have applied 3-D $5 \times 5 \times 5$ masked CNN[8] to extract spatial priors, which are fused with hyperpriors in PA for entropy context modeling (e.g., the bottom part of Fig. 9). Here, we have assumed the single Gaussian distribution for the context modeling of entropy coding. Note that temporal priors are not used for intrapixel and interresidual in this article by only utilizing the spatial priors.

The original NLAIC applies multiple NLAMs in both main and hypercoders, leading to excessive memory consumption at a large spatial scale. In E2E-NVC, NLAMs are only used at the bottleneck layers for both main and hyper-encoder–decoder pairs, allowing bits to be allocated adaptively.

To overcome the nondifferentiability of the quantization operation, quantization is usually simulated by adding uniform noise in [142]. However, such noise augmentation is not exactly consistent with the rounding in inference, which can yield performance loss, as reported by [135]. Thus, we apply universal quantization (UQ) [135] in neuro-Intra. UQ is used for neuro-Motion and neuro-Res

[7]Intracoding and residual coding only use joint spatial and hyperpriors without temporal inference.

[8]This $5 \times 5 \times 5$ convolutional kernel shares the same parameters for all channels, offering great model complexity reduction compared with the 2-D CNN-based solution in [148].

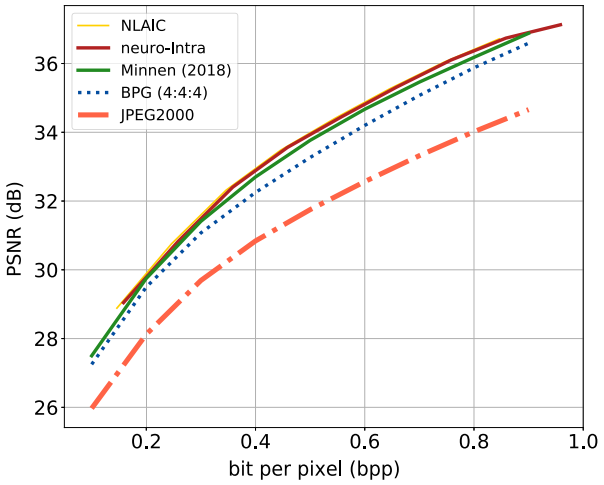

Fig. 7. ***Efficiency of neuro-Intra. PSNR versus rate performance of neuro-Intra in comparison to NLAIC [136], Minnen (2018) [148], BPG (4:4:4), and JPEG2000. Note that the curves for neuro-Intra and NLAIC overlap.***

as well. When applied to the common Kodak data set, neuro-Intra performs NLAIC [136] and outperforms Minnen (2018) [148], BPG (4:4:4), and JPEG2000, as shown in Fig. 7.

C. Neural Motion Coding and Compensation

Interframe coding plays a vital role in video coding. The key is how to efficiently represent motion in a compact format for compensation. In contrast to the pixel-domain block-based MEMC in conventional video coding, we rely on optical flow to accurately capture the temporal information for *motion compensation*.

To improve interframe prediction, we extend our earlier work [131] to multiscale motion generation and compensation. This multiscale motion processing directly transforms two concatenated frames (where one frame is the reference from the past, and one is the current frame) into quantized temporal features that represent the interframe motion. These quantized features are decoded into compressed optical flow in an unsupervised way for frame compensation via warping. This one-stage scheme does not require any pretrained flow network, such as FlowNet2 or PWC-net, to generate the optical flow explicitly. It allows us to quantize the motion features rather than the optical flow and to train the motion feature encoder and decoder together with explicit consideration of quantization and rate constraint.

The neuro-Motion module is modified for multiscale motion generation, where the main encoder is used for feature fusion. We replace the main decoder with a *pyramidal flow decoder*, which generates the multiscale compressed optical flows (MCFs). MCFs will be processed together with the reference frame, using an *MS-MCN* to obtain the predicted frame efficiently, as shown in Fig. 8. Refer to [216] for more details.

Encoding motion compactly is another important factor for overall performance improvement. We suggest the joint spatiotemporal and hyperprior-based context-adaptive model shown in Fig. 9 for efficiently inferring current quantized features. This is implemented in the PA engine of Fig. 6(b).

The joint spatiotemporal and hyperprior-based context-adaptive model mainly consists of a *spatiotemporal-hyperaggregation module* (STHAM) and a *temporal updating module* (TUM), as shown in Fig. 9. At timestamp t, STHAM is introduced to accumulate all the accessible priors and estimate the mean and standard deviation of GMM jointly using

$$(\mu_{\mathscr{F}}, \sigma_{\mathscr{F}}) = \mathbb{F}(\mathscr{F}_1, \ldots, \mathscr{F}_{i-1}, \hat{\mathbf{z}}_t, \mathbf{h}_{t-1}). \tag{1}$$

Spatial priors are autoregressively derived using masked 5×5×5 3-D convolutions and then concatenated with decoded hyperpriors and temporal priors using stacked 1×1×1 convolutions. $\mathscr{F}_i, i = 0, 1, 2, \ldots$ are elements of quantized latent features (e.g., motion flow). $\mathbf{h}_{t-1}$ is aggregated temporal priors from motion flows preceding the current frame. The neuro-Motion module exploits temporal redundancy to further predict the efficiency, leveraging the correlation between second-order moments of intermotion. A probabilistic model of each element to be encoded is derived with the estimated $\mu_{\mathscr{F}}$ and $\sigma_{\mathscr{F}}$ by

$$p_{\mathscr{F}|(\mathscr{F}_1,\ldots,\mathscr{F}i-1,\hat{\mathbf{z}}_t,\,\mathbf{h}_{t-1})}(\mathscr{F}_i|\mathscr{F}_1,\ldots,\mathscr{F}_{i-1},\hat{\mathbf{z}}_t,\mathbf{h}_{t-1}) = \prod_i \left(\mathcal{N}(\mu_{\mathscr{F}},\sigma^2_{\mathscr{F}}) * \mathcal{U}\left(-\frac{1}{2},\frac{1}{2}\right)\right)(\mathscr{F}_i). \tag{2}$$

Note that TUM is applied to embed current quantized features $\mathscr{F}_t$ recurrently using a standard ConvLSTM [220]

$$(\mathbf{h}_t, \mathbf{c}_t) = \mathrm{ConvLSTM}(\mathscr{F}_t, \mathbf{h}_{t-1}, \mathbf{c}_{t-1}) \tag{3}$$

where $\mathbf{h}_t$ are updated temporal priors for the next frame, and $\mathbf{c}_t$ is a memory state to control information flow across multiple time instances (e.g., frames). Other recurrent units can also be used to capture temporal correlations as in (3).

It is worth noting that leveraging second-order information for the representation of compact motion is also widely explored in traditional video coding approaches. For example, motion vector predictions from spatial and temporal colocated neighbors are standardized in H.265/HEVC, by which only motion vector differences (after prediction) are encoded.

D. Neural Residual Coding

Interframe residual coding is another significant module contributing to the overall efficiency of the system. It is used to compress the temporal prediction error pixels.

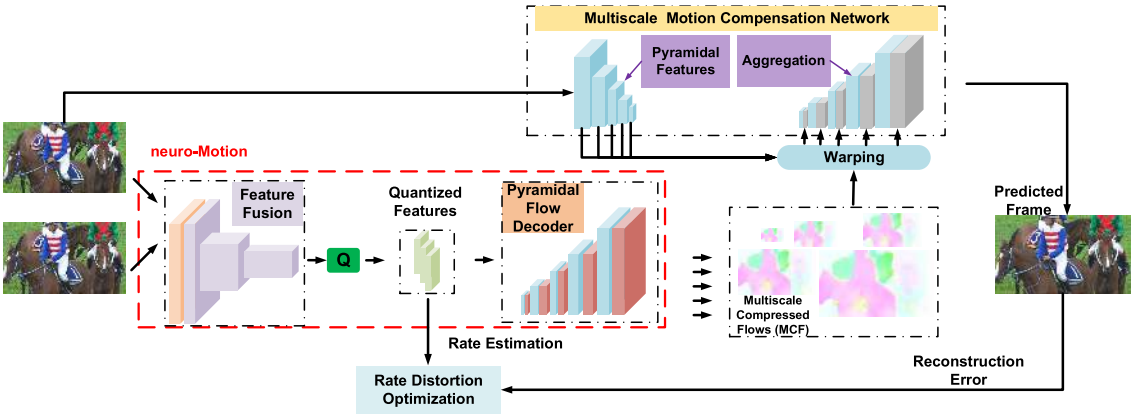

Fig. 8. ***Multiscale MEMC. One-stage neuro-Motion with MS-MCN uses a pyramidal flow decoder to synthesize the MCFs that are used in an MS-MCN for generating predicted frames.***

It affects the efficiency of next frame prediction since errors usually propagate temporally.

Here, we use the VAE architecture in Fig. 6(b) to encode the residual $\mathbf{r}_t$. The rate-constrained loss function is used

$$L = \lambda \cdot \mathbb{D}_2\left(\mathbf{X}_t, \left(\mathbf{X}_t^p + \hat{\mathbf{r}}_t\right)\right) + R \tag{4}$$

where $\mathbb{D}_2$ is the ℓ_2 loss between a residual compensated frame $\mathbf{X}_t^p + \hat{\mathbf{r}}_t$ and $\mathbf{X}_t$. neuro-Res will be first pretrained using the frames predicted by the pretrained neuro-Motion and MS-MCN and a loss function in (4) where the rate R only accounts for the bits for residual. Then, we refine neuro-Res jointly with neuro-Motion and MS-MCN, using a loss where R incorporates the bits for both motion and residual with two frames.

E. Experimental Comparison

We apply the same low-delay coding setting as DVC in [129] for our method, the H.264/AVC, and H.265/HEVC for comparison. We encode 100 frames and use a GoP size of 10 on H.265/HEVC test sequences and 600 frames with a GoP size of 12 on the UVG data set. For the H.265/HEVC, we apply the fast mode of the $\times 265$[9]—a popular open-source H.265/HEVC encoder implementation—while the fast mode of the $\times 264$[10] is used as the representative of the H.264/AVC encoder.

We show the leading compression efficiency in Fig. 10 using respective PSNR and MS-SSIM measures, across H.265/HEVC and UVG test sequences. In Table 3, by setting the same anchor using the H.264/AVC, our NVC presents 35% BD-Rate gains, while H.265/HEVC and DVC offer 30% and 22% gains, respectively. If the distortion is measured by MS-SSIM, our gains in efficiency are even larger. This demonstrates that NVC can achieve a 50% improvement in efficiency, while both the H.265/HEVC and DVC achieve only around 25%.

Our NVC rivals the recent DVC_Pro [221], an upgrade of the earlier DVC [141], for example, 35.54% and 50.83% BD-Rate reduction measured by PSNR and MS-SSIM distortion, respectively, for NVC, while 34.57% and 45.88% are marked for DVC_Pro. DVC [141] has mainly achieved a higher level of coding efficiency than the H.265/HEVC at high bit rates. However, a sharp decline in the performance of DVC is revealed at low bit rates (e.g., performing worse than the H.264/AVC at some rates). We have also observed that DVC's performance varies for different test sequences. DVC_Pro upgrades DVC with better intracoding/residual coding using [148] and λ fine-tuning, showing state-of-the-art performance [221].

Visual comparison. We provide a visual quality comparison between NVC, the H.264/AVC, and H.265/HEVC, as shown in Fig. 11. Generally, NVC yields reconstructions that are much higher in quality than those of its competitors, even with a lower bit rate cost. For the sample clip "RaceHorse," which includes nontranslational motion and complex background, NVC uses 7% fewer bits despite an improvement in quality greater than 1.5-dB

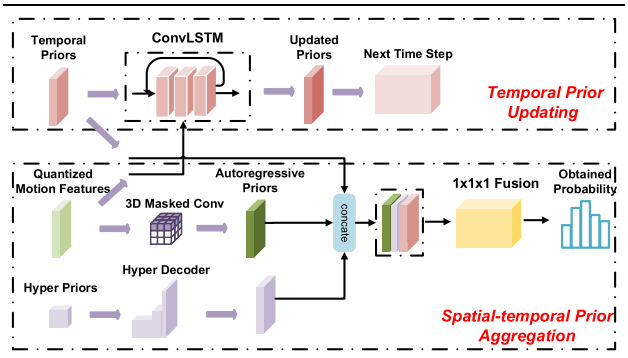

Fig. 9. ***Context-adaptive modeling using joint spatiotemporal and hyperpriors. All priors are fused in PA to provide estimates of the probability distribution parameters.***

[9]http://x265.org/
[10]https://www.videolan.org/developers/x264.html

Table 3 BD-Rate Gains of NVC, H.265/HEVC, and DVC Against the H.264/AVC

Sequences	H.265/HEVC				DVC				NVC			
	PSNR		MS-SSIM		PSNR		MS-SSIM		PSNR		MS-SSIM	
	BDBR	BD-(D)	BDBR	BD-(D)	BDBR	BD-(D)	BDBR	BD-(D)	BDBR	BD-(D)	BDBR	BD-(D)
ClassB	-32.03%	0.78	-27.67%	0.0046	-27.92%	0.72	-22.56%	0.0049	**-45.66%**	**1.21**	**-54.90%**	**0.0114**
ClassC	-20.88%	0.91	-19.57%	0.0054	-3.53%	0.13	-24.89%	0.0081	-17.82%	**0.73**	**-43.11%**	**0.0133**
ClassD	-12.39%	0.57	-9.68%	0.0023	-6.20%	0.26	-22.44%	0.0067	**-15.53%**	**0.70**	**-43.64%**	**0.0123**
ClassE	-36.45%	0.99	-30.82%	0.0018	-35.94%	1.17	-29.08%	0.0027	**-49.81%**	**1.70**	**-58.63%**	**0.0048**
UVG	-48.53%	1.00	-37.5%	0.0056	-37.74%	1.00	-16.46%	0.0032	**-48.91%**	**1.24**	**-53.87%**	**0.0100**
Average	-30.05%	0.85	-25.04%	0.0039	-22.26%	0.65	-23.08%	0.0051	**-35.54%**	**1.11**	**-50.83%**	**0.0103**

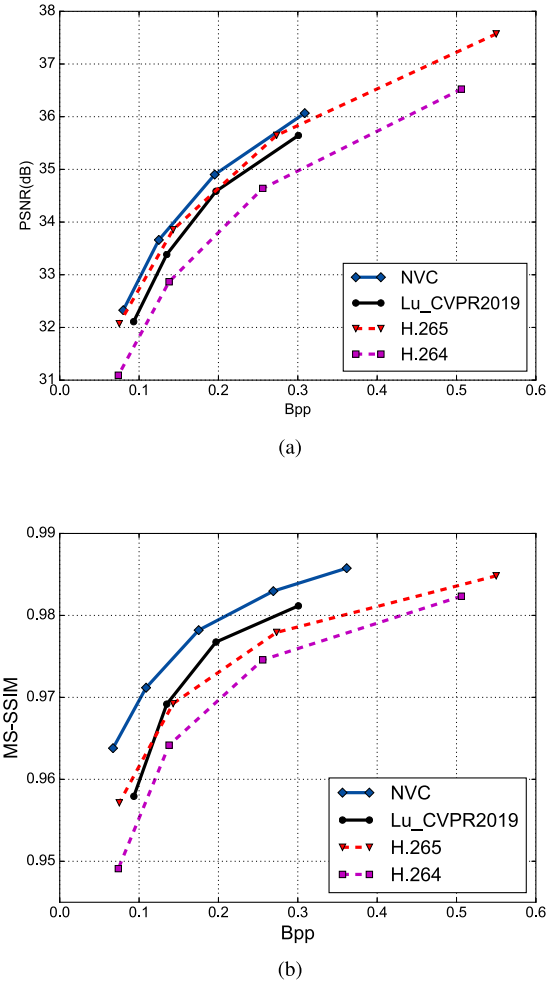

Fig. 10. ***BD-Rate illustration using PSNR & MS-SSIM. (a) NVC offers averaged 35.34% gain against the anchor H.264/AVC when distortion is measured using PSNR. (b) NVC shows over 50% gains against the anchor H.264/AVC when using MS-SSIM evaluation. MS-SSIM is usually studied as a perceptual quality metric in image compression, especially at a low bit rate.***

PSNR compared with the H.264/AVC. For other cases, our method also shows robust improvement. Traditional codec usually suffers from blocky artifacts and motion-induced noise close to the edges of objects. In the H.264/AVC, you clearly can observe block partition boundaries with severe pixel discontinuity. Our results provide higher quality reconstruction and avoid noise and artifacts.

F. Discussion and Future Direction

We developed an end-to-end deep neural video coding framework that can learn a compact spatiotemporal representation of raw video input. Our extensive simulations yield very encouraging results, demonstrating that our proposed method can offer consistent and stable gains over existing methods (e.g., the traditional H.265/HEVC and recent learning-based approaches [129]) across a variety of bit rates and a wide range of contents.

The H.264/AVC, H.264/HEVC, AVS, AV1, and even the VVC are masterpieces of hybrid prediction/transform framework-based video coding. R-D optimization, rate control, and so on can certainly be incorporated to improve learning-based solutions. For example, reference frame selection is an important means by which we can embed and aggregate the most appropriate information for reducing temporal error and improving overall intercoding efficiency. Making deep learning-based video coding practically applicable is another direction worthy of deeper investigation.

VII. CASE STUDIES FOR POSTPROCESSING: EFFICIENT NEURAL FILTERING

In this case study, both in-loop and postfiltering are demonstrated using stacked DNN-based neural filters for quality enhancement of reconstructed frames. We specifically design a single-frame guided CNN that adapts pretrained CNN models to different video contents for in-loop filtering and a multiframe CNN leveraging spatiotemporal information for postfiltering. Both reveal noticeable performance gains. In practice, neural filters can be devised, that is, in-loop or post, according to the application requirements.

A. In-Loop Filtering via Guided CNN

As reviewed in Section IV, most existing works design a CNN model to directly map a degraded input frame to its restored version (e.g., ground-truth label), as illustrated in Fig. 12(a). To ensure that the model is generalizable to other contexts, CNN models are often designed to use deeper layers, denser connections, wider receptive fields, and so on, with hundreds of millions of parameters. As a consequence, such generalized models are poorly suited to most practical applications. To address this problem, we propose that content-adaptive weights can be used

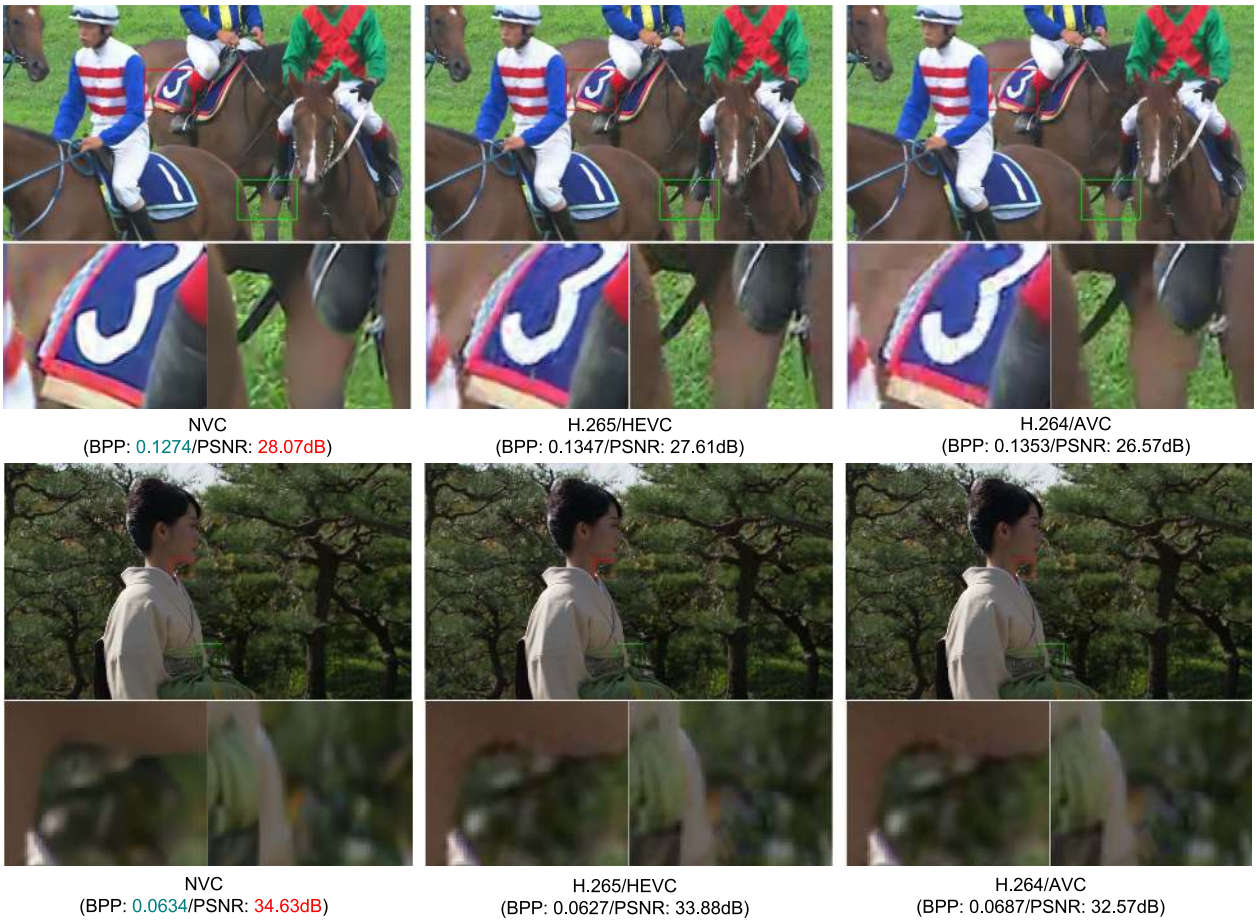

Fig. 11. ***Visual comparison. Reconstructed frames of NVC, H.265/HEVC, and H.264/AVC. We avoid blocky artifacts, visible noise, and so on and provide better quality at lower bit rate.***

to guide a shallow CNN model [as shown in Fig. 12(b)] instead.

The principle underlying this approach is sparse signal decomposition: we expect that the CNN model can represent any input as a weighted combination of channelwise features. Note that weighting coefficients are dependent on input signals, making this model generalizable to a variety of content characteristics.

1) Method: Let $\mathbf{x}$ be a degraded block with N pixels in a columnwise vector format. The corresponding source block of $\mathbf{x}$ is $\mathbf{s}$, which has a processing error $\mathbf{d} = \mathbf{s} - \mathbf{x}$. We wish to have $\mathbf{r}_{\text{corr}}$ from $\mathbf{x}$ so that the final reconstruction $\mathbf{x}_{\text{corr}} = \mathbf{x} + \mathbf{r}_{\text{corr}}$ is closer to $\mathbf{s}$.

Let the CNN output layer have M channels, that is, $\mathbf{r}_0$, $\mathbf{r}_1, \ldots, \mathbf{r}_{M-1}$. Then, $\mathbf{r}_{\text{corr}}$ is assumed as a linear combination of these channelwise feature vectors

$$\mathbf{r}_{\text{corr}} = a_0\mathbf{r}_0 + a_1\mathbf{r}_1 + \cdots + a_{M-1}\mathbf{r}_{M-1} \tag{5}$$

where $a_0, a_1, \ldots, a_{M-1}$ are the weighting parameters that are explicitly signaled in the compressed bitstream.

Our objective is to minimize the distance between the restored block $\mathbf{x}_{\text{corr}}$ and its corresponding source $\mathbf{s}$, that is, $|\mathbf{x}_{\text{corr}} - \mathbf{s}|^2 = |\mathbf{r}_{\text{corr}} - \mathbf{d}|^2$. Given the channelwise

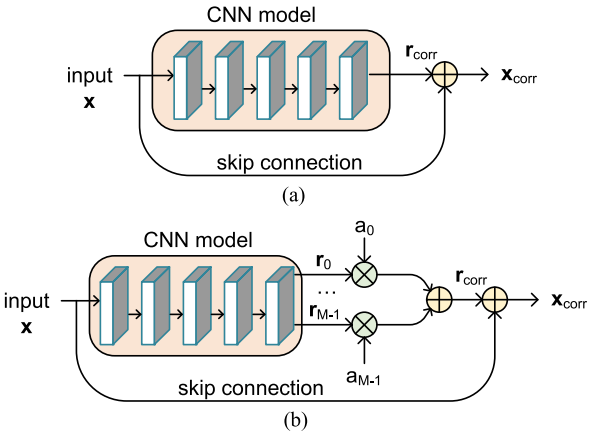

Fig. 12. ***CNN-based restoration. (a) Conventional model structure. (b) Guided CNN model with adaptive weights.***

Table 4 Layered Structure and Parameter Settings of the CNN Model Used

Layer	Kernel size	Input channels	Output channels	Parameters
1	3×3	1	16	144
2	3×3	16	8	1152
3	3×3	8	8	576
4	3×3	8	8	576
5	3×3	8	8	576
6	3×3	8	8	576
7	3×3	8	M	144
Total parameters				3744

output features $\mathbf{r}_0$, $\mathbf{r}_1, \ldots, \mathbf{r}_{M-1}$, for a degraded input $\mathbf{x}$, the weighting parameters $a_0, a_1, \ldots, a_{M-1}$ can then be estimated by least-squares optimization as

$$[a_0, a_1, \ldots, a_{M-1}]^{\mathrm{T}} = (\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{d} \tag{6}$$

where $\mathbf{R} = [\mathbf{r}_0, \mathbf{r}_1, \ldots, \mathbf{r}_{M-1}]$ is the matrix at a size of $N\times M$ comprised of stacked output features in columnwise order. The reconstruction error is given by

$$e = |\mathbf{r}_{\mathrm{corr}} - \mathbf{d}|^2 = |\mathbf{d}|^2 - \mathbf{d}^{\mathrm{T}}\mathbf{R}(\mathbf{R}^{\mathrm{T}}\mathbf{R})^{-1}\mathbf{R}^{\mathrm{T}}\mathbf{d}. \tag{7}$$

2) Loss Function: Assuming that one training batch is comprised of T patch pairs: $\{\mathbf{s}_i, \mathbf{x}_i\}, i = 0, 1, \ldots, \mathrm{T}-1$, the overall reconstruction error over the training set is

$$E = \sum_i \left\{|\mathbf{d}_i|^2 - \mathbf{d}_i{}^{\mathrm{T}}\mathbf{R}_i\left(\mathbf{R}_i{}^{\mathrm{T}}\mathbf{R}_i\right)^{-1}\mathbf{R}_i{}^{\mathrm{T}}\mathbf{d}_i\right\} \tag{8}$$

where $\mathbf{d}_i = \mathbf{s}_i - \mathbf{x}_i$ is the error for the ith patch. $\mathbf{R}_i = [\mathbf{r}_{i,0}, \mathbf{r}_{i,1}, \ldots, \mathbf{r}_{i,M-1}]$ is the corresponding channelwise features in matrix form, with $\mathbf{r}_{i,j}$ being the jth channel when the training sample $\mathbf{x_i}$ is passed through the CNN model. Given that $|\mathbf{d}_i|^2$ is independent of the network model, the loss function can be simplified as

$$L = \sum_i \left\{-\mathbf{d}_i{}^{\mathrm{T}}\mathbf{R}_i\left(\mathbf{R}_i{}^{\mathrm{T}}\mathbf{R}_i\right)^{-1}\mathbf{R}_i{}^{\mathrm{T}}\mathbf{d}_i\right\}. \tag{9}$$

3) Experimental Studies: A shallow baseline CNN model (as described in Table 4) is used to demonstrate the efficiency of the guided CNN model. This model is comprised of seven layers in total and has a fixed kernel size of 3×3. At the bottleneck layer, the channel number of the output feature map is M. After extensive simulations, $M = 2$ was selected. In total, our model only requires 3744 parameters, far fewer than the number required by existing methods.

In training, 1000 pictures of DIV2K [222] are used. All frames are compressed using the AV1 encoder with in-loop filters CDEF [158] and LR [159] turned off to generate corresponding quantization-induced degraded reconstructions. We divide the 64 QPs into six ranges and trained one model for each QP range. The six ranges include QP values 7–16, 17–26, 27–36, 47–56, and 57–63. Compressed frames falling into the same QP range are used to train the corresponding CNN model. Frames are segmented into 64×64 patches. Each batch contains 1000 patches. We adopt the Adaptive moment estimation (Adam) algorithm, with the initial learning rate set at 1e-4. The learning rate is halved every 20 epochs.

We use the Tensorflow platform, which runs on NVIDIA GeForce GTX 1080Ti GPU, to evaluate coding efficiency across four QPs, for example, {32, 43, 53, and 63}. Our test set includes 24 video sequences with resolutions ranging from 2560×1600 to 352×288. The first 50 frames of each sequence are tested in both intraconfiguration and interconfiguration.

In our experiments, N is set to 64, 128, 256, and the whole frame, respectively. We find that $N = 256$ yields the best performance. For each block, the linear combination parameters a_i $(i = 0, 1)$ are derived accordingly. To strike an appropriate balance between bit consumption and model efficiency, our experiments suggest that the dynamic range of a_i is within 15.

We compare the respective BD-Rate reductions of our guided CNN model and a baseline CNN model against the AV1 baseline encoder. All filters are enabled for the AV1 anchor. For a description of the baseline CNN model, see Table 4 with $M = 1$. Our guided CNN model is the baseline model with $M = 2$ plus the adaptive weights.

Both baseline and guided CNN models are applied on top of the AV1 encoder with only the deblocking filter enabled and other filters (including CDEF and LR) turned off. The findings reported in Table 5 demonstrate that either baseline or guided CNN models can be used to replace additional adaptive in-loop filters while improving the R-D efficiency. Furthermore, regardless of the block size and frame types, our guided model always outperforms the baseline CNN. This is mainly due to the adaptive weights used to better characterize content dynamics. Similar lightweight CNN structures can be upgraded using deep models [162], [163], [166] for potentially greater BD-Rate savings.

B. Multiframe Postfiltering

This section demonstrates how multiframe video enhancement (MVE) scheme-based postfiltering can be used to minimize compression artifacts. We implement our proposed approach on AV1 reconstructed frames and achieve significant coding improvement. Similar observations are expected with different anchors, such as the H.265/HEVC.

1) Method: Single-frame video enhancement (SVE) refers to the sole application of the fusion network without leveraging temporal frame correlations. As discussed in Section IV, there are a great number of network models that can be used to do SVE. In most cases, the efficiency and complexity are at odds with one another: in other words, efficiency and complexity come at the cost of deeper networks and higher numbers of parameters.

Table 5 BD-Rate Savings of Baseline and Guided CNN Models Against the AV1 Baseline

Resolution	Sequence	All Intra					Random Access				
		Baseline CNN	Guided CNN N=64	Guided CNN N=128	Guided CNN N=256	Guided CNN Frame	Baseline CNN	Guided CNN N=64	Guided CNN N=128	Guided CNN N=256	Guided CNN Frame
2560 × 1600	PeopleOnStreet	−1.15%	−1.95%	−2.84%	−2.90%	−2.81%	−0.19%	−0.22%	−1.03%	−1.02%	−0.83%
	Traffic	−1.71%	−1.76%	−3.01%	−3.16%	−3.03%	−0.26%	+1.89%	−1.64%	−2.15%	−2.17%
1920 × 1080	BasketballDrive	−0.45%	+2.95%	−0.72%	−1.06%	−0.72%	−0.02%	+8.04%	+0.87%	+0.07%	−0.05%
	BQTerrace	−0.98%	−3.19%	−3.66%	−3.44%	−2.10%	−0.33%	+0.68%	−1.62%	−1.91%	−1.51%
	Cactus	−1.64%	−1.38%	−2.79%	−2.89%	−2.56%	−0.21%	+1.18%	−1.13%	−1.31%	−0.96%
	Kimono	−0.23%	+3.55%	−0.18%	−0.88%	−0.95%	−0.07%	+6.07%	+0.84%	−0.07%	−0.01%
	ParkScene	−1.21%	+0.01%	−1.92%	−2.21%	−2.11%	−0.07%	+1.11%	−1.46%	−1.82%	−0.92%
	blue-sky	−2.89%	−0.96%	−2.58%	−2.86%	−2.56%	+0.00%	+3.46%	−2.02%	−2.96%	−2.77%
	crowd_run	−3.01%	−2.34%	−3.11%	−3.22%	−3.08%	−0.13%	−1.69%	−2.19%	−2.07%	−1.09%
832 × 480	BasketballDrill	−2.99%	−5.55%	−6.45%	−6.26%	−5.88%	−0.25%	−0.33%	−2.10%	−1.79%	−1.55%
	BQMall	−1.74%	−3.96%	−4.48%	−4.46%	−4.35%	−0.15%	+0.16%	−1.05%	−1.13%	−0.76%
	PartyScene	−0.83%	−3.77%	−4.02%	−3.97%	−3.81%	−0.20%	−1.10%	−1.43%	−1.25%	−0.13%
	RaceHorsesC	−1.91%	−2.01%	−2.58%	−2.49%	−2.38%	−0.21%	−0.70%	−1.28%	−1.03%	−0.80%
416 × 240	BasketballPass	−3.08%	−3.66%	−4.60%	−4.72%	−4.65%	−0.20%	+0.71%	−0.63%	−0.62%	−0.36%
	BlowingBubbles	−2.60%	−3.36%	−3.78%	−3.77%	−3.76%	−0.34%	−0.55%	−1.05%	−0.87%	−0.86%
	BQSquare	−4.92%	−6.09%	−6.23%	−6.27%	−6.22%	−0.50%	−0.54%	−0.92%	−1.13%	−1.17%
	RaceHorses	−3.57%	−5.39%	−5.75%	−5.75%	−5.76%	−0.51%	−2.82%	−3.06%	−2.69%	−2.94%
1280 × 720	Johnny	−2.01%	−2.41%	−4.03%	−4.21%	−4.12%	−0.31%	+8.32%	−0.94%	−2.57%	−2.63%
	FourPeople	−1.94%	−0.54%	−3.49%	−3.76%	−2.85%	−0.29%	+17.99%	+1.20%	−1.65%	−1.60%
	KristenAndSara	−2.71%	−1.49%	−3.97%	−4.32%	−4.26%	−0.42%	+15.95%	+0.53%	−2.49%	−2.31%
352 × 288	Harbour	−0.79%	−1.18%	−1.43%	−1.38%	−1.42%	−0.23%	−1.00%	−1.29%	−1.40%	−1.08%
	Ice	−3.59%	−5.54%	−6.88%	−7.08%	−7.19%	−0.59%	−1.59%	−3.59%	−3.65%	−3.97%
	Silent	−1.68%	−1.88%	−2.80%	−2.77%	−2.79%	−0.21%	+1.96%	−0.29%	−0.27%	−0.70%
	Students	−3.08%	−4.10%	−4.77%	−4.81%	−4.88%	−0.52%	+1.25%	−1.16%	−1.44%	−1.66%
	Average	−2.11%	−2.33%	−3.59%	−3.69%	−3.51%	−0.26%	+2.43%	−1.10%	−1.55%	−1.37%

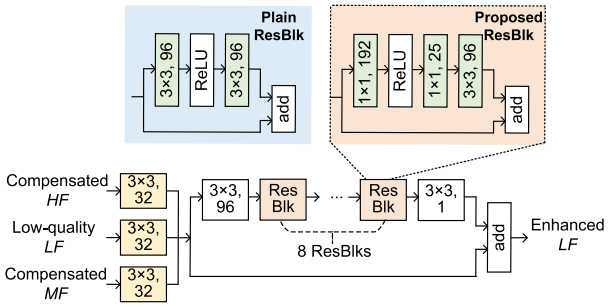

Fig. 13. ***WARN. This WARN is used to fuse/enhance the input frame for improved quality. In the MVE case, it takes three inputs to enhance the LFs; in the SVE case, it inputs a single frame and outputs its enhanced version. This WARN generally follows the ResNet structure with the residual link and ResBlk embedded. Note that ResBlk is extended to support wide activation from its plain version prior to ReLU activation.***

Recently, Yu *et al.* [223] discovered that models with more feature channels before activation could provide significantly better performance with the same parameters and computational budgets. We design a WARN by combining wide activation with a powerful deep residual network (ResNet) [224], as shown in Fig. 13. This WARN illustrates the three inputs for an enhanced output in the MVE framework. In contrast, SVE normally inputs a single frame and outputs a corresponding enhanced representation.

This MVE closely follows the two-step strategy reviewed in Section IV. It uses FlowNet2 [186] to perform pixel-level motion estimation-/compensation-based temporal frame alignment. Next, a WARN-based fusion network is used for final enhancement. We allow the two high-quality frames (HFs) immediately preceding and succeeding a low-quality frame (LF) to enhance the LF in between. Bidirectional warping is performed for each LF to produce compensated HFs in Fig. 14.

2) Experimental Studies: We evaluate both SVE and MVE against the AV1 baseline. A total of 118 video sequences were selected to train network models. More specifically, the first 200 frames of each sequence are encoded with the AV1 encoder to generate the reconstructed frames. The QPs are {32, 43, 53, 63}, yielding 23 600 reconstructed frames in total. After frame alignment, we select one training set containing compensated HF_0, compensated HF_1, and to-be-enhanced LF from every eight frames, yielding a total of 2900 training sets. These sets are used to train the WARN model as the fusion network. Notice that we train the WARN models for SVE and MVE individually. The GoP size is 16 with a hierarchical prediction structure. The LFs and HFs are identified using their QPs, that is, HFs with lower QP than the base QP are decoded, such as frames 0, 4, 8, 12, and 16 in Fig. 15.

Algorithms are implemented using the Tensorflow platform, NVIDIA GeForce GTX 1080Ti GPU. In training, frames are segmented into 64×64 patches, with 64 patches included in each batch. We adopt the Adam optimizer with the initial learning rate set at 1e-4. The learning rate can then be adjusted using the step strategy with $\gamma = 0.5$. Additional 18 sequences are also employed for testing. The first 50 frames of each test sequence are compressed. Then, the reconstructed frames are enhanced using the proposed SVE and MVE methods.

We apply the proposed method to AV1 reconstructed frames. The results are presented in Table 6. Due to the

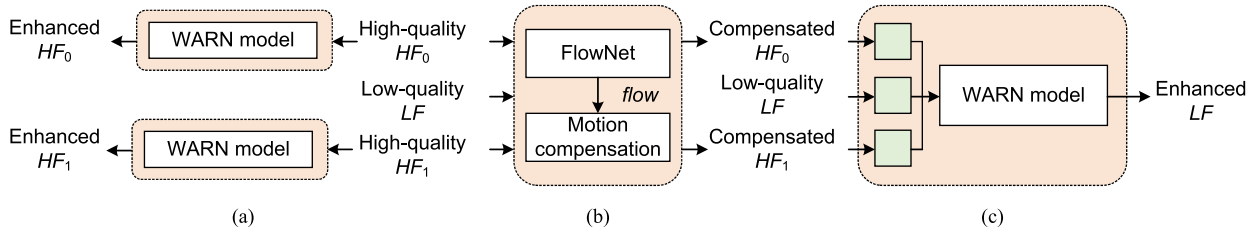

Fig. 14. ***Enhancement framework. (a) Single-input WARN-based SVE to enhance the HF. (b) and (c) Two-step MVE using FlowNet2 for temporal alignment and three-input WARN-based fusion to use preceding and succeeding HFs for LF enhancement.***

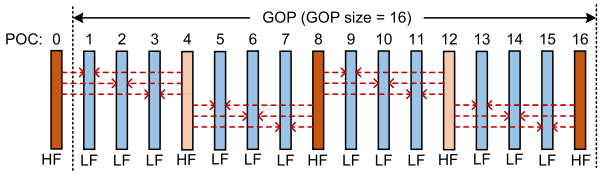

Fig. 15. ***Hierarchical coding structure in the AV1 encoder. The LFs are enhanced using HFs following the prediction structure via MVE scheme, and HFs are restored using SVE method.***

Table 6 BD-Rate Improvement of the Proposed SVE and MVE Scheme Against the AV1

Class	Sequence	All Intra		Random Access	
		SVE	MVE	SVE	MVE
A	PeopleOnStreet	−9.1%	−14.7%	−5.0%	−8.1%
	Traffic	−7.6%	−22.2%	−5.8%	−8.8%
B	BasketballDrive	−5.9%	−13.1%	−4.4%	−6.4%
	BQTerrace	−8.0%	−23.7%	−7.7%	−9.8%
	Cactus	−7.7%	−21.9%	−3.9%	−6.0%
	Kimono	−3.8%	−20.4%	−3.9%	−7.1%
	ParkScene	−5.1%	−26.3%	−4.9%	−8.0%
C	BasketballDrill	−12.5%	−21.3%	−5.6%	−7.9%
	BQMall	−8.9%	−18.7%	−3.5%	−6.1%
	PartyScene	−7.2%	−19.0%	−3.2%	−5.0%
	RaceHorsesC	−5.9%	−18.3%	−3.3%	−5.6%
D	BasketballPass	−10.0%	−18.5%	−3.4%	−6.2%
	BlowingBubbles	−7.0%	−19.8%	−4.6%	−6.7%
	BQSquare	−10.8%	−21.3%	−11.0%	−13.6%
	RaceHorses	−9.2%	−19.3%	−4.9%	−7.8%
E	FourPeople	−9.7%	−21.7%	−5.1%	−7.4%
	Johnny	−9.6%	−20.7%	−5.5%	−8.0%
	KristenAndSara	−9.6%	−21.2%	−4.4%	−7.0%
	Average	−8.2%	−20.1%	−5.0%	−7.5%

hierarchical coding structure in interprediction, the LFs in Fig. 15 are enhanced using the neighboring HFs via the MVE framework. The HFs themselves are enhanced using the SVE method.

The overall BD-Rate savings of the SVE and MVE methods are tabulated in Table 6, against the AV1 baseline. SVE achieves an averaged reduction of 8.2% and 5.0% BD-Rate for all intra and random access scenarios, respectively. On the other hand, our MVE obtains 20.1% and 7.5% BD-Rate savings on average, further demonstrating the effectiveness of our proposed scheme. When random access techniques are used, the HFs selected are generally distant from a target LF, which reduces the benefits provided from inter-HFs. On the other hand, intracoding techniques uniformly demonstrate greater BD-Rate savings because the neighboring frames nearest to target LFs can be used. This contributes significantly to enhancement.

Besides the objective measures, sample snapshots of reconstructed frames are illustrated in Fig. 16, clearly demonstrating that blocky and ringing artifacts from the AV1 baseline are attenuated after applying either SVE or MVE-based filtering. Notably, MVE creates more visually appealing images than SVE.

C. Discussion and Future Direction

In this section, we propose DNN-based approaches for video quality enhancement. For in-loop filtering, we develop a guided CNN framework to adapt pretrained CNN models to various video contents. Under this framework, the guided CNN learns to project an input signal onto a subspace of dimension M. The weighting parameters for a linear combination of these channels are explicitly signaled in the encoded bitstream to obtain the final restoration. For postfiltering, we devise a spatiotemporal multiframe architecture to alleviate the compression artifacts. A two-step scheme is adopted in which optical flow is first obtained for accurate motion estimation/compensation, and then, a WARN is designed for information fusion and quality enhancement. Our proposed enhancement approaches can be implemented on different CNN architectures.

The quality of enhanced frames plays a significant role in overall coding performance since they serve as reference

Fig. 16. ***Qualitative visualization. Zoomed-in snapshots of reconstructed frames for the AV1 baseline, SVE and MVE filtered restoration, and the ground-truth label.***

frames for the motion estimation of subsequent frames. Our future work will investigate the joint effect of in-loop filtering and motion estimation on reference frames to exploit the inherent correlations of these coding tools, which could further improve coding performance.

VIII. CONCLUSION AND DISCUSSION

As an old Chinese saying goes, "a journey of a thousand miles begins with a single step." This is particularly true in the realm of technological advancement. Both the fields of video compression and machine learning have been established for many decades, but, until recently, they evolved separately in both academic explorations and industrial practice.

Lately, however, we have begun to witness the interdisciplinary advancements yielded by the proactive application of deep learning technologies [225] into video compression systems. The benefits of these advances include remarkable improvements in performance in many technical aspects. To showcase the remarkable products of this disciplinary cross-pollination, we have identified three major functional blocks in a practical video system, for example, preprocessing, coding, and postprocessing. We then reviewed related studies and publications to help the audience familiarize themselves with these topics. Finally, we presented three case studies to highlight the state-of-the-art efficiency resulting from the application of DNNs to video compression systems, which demonstrates this avenue of exploration's great potential to bring about a new generation of video techniques, standards, and products.

Though this article presents separate DNN-based case studies for preprocessing, coding, and postprocessing, we believe that a fully end-to-end DNN model could potentially offer a greater improvement in performance while enabling more functionalities. For example, Xia *et al.* [226] applied deep object segmentation in preprocessing and used it to guide neural video coding, demonstrating noticeable visual improvements at very low bit rates. Meanwhile, Lee *et al.* [152] and others observed similar effects when a neural adaptive filter was successfully used to further enhance neural compressed images.

Nevertheless, a number of open problems requiring substantial further study have been discovered. These include the following.

1) *Model generalization.* It is vital for DNN models to be generalizable to a wide variety of video content, different artifacts, and so on. Currently, most DNN-based video compression techniques utilize supervised learning, which often demands a significant amount of labeled image/video data for the full spectrum coverage of the aforementioned application scenarios. Continuously developing a large-scale data set, such as the ImageNet,[11] presents one possible solution to this problem. An alternative approach may use more advanced techniques to alleviate uncertainty related to a limited training sample for model generalization. These techniques include (but are not limited to) few-shot learning [227] and self-supervised learning [225].
2) *Complexity.* Existing DNN-based methods are mainly criticized for their unbearable complexity in both computational and spatial dimensions. Compared to conventional video codecs which require tens of kilobytes of on-chip memory, most DNN algorithms require several megabytes or even gigabytes of memory space. On the other hand, although inference may be very fast, training could take hours, days, or even weeks for converged and reliable models [141]. All of these issues present serious barriers to the market adoption of DNN-based tools, particularly on energy-efficient mobile platforms. One promising solution is to design specialized hardware for the acceleration of DNN algorithms [157]. Currently, neural processing units (NPUs) have attracted significant attention and have been gradually deployed in heterogeneous platforms (e.g., Qualcomm AI Engine in the Snapdragon chip series and Neural Processor in Apple silicon). This paints a promising picture of a future in which DNN algorithms can be deployed on NPU-equipped devices at a massive scale.
3) *QoE metric.* Video quality matters. A video QoE metric that is better correlated with the HVS is highly desirable not only for quality evaluation, but also for loss control in DNN-based video compression. There has been a notable development in both subjective and objective video quality assessments, yielding several well-known metrics, such as SSIM [228], just-noticeable-distortion (JND) [229], and VMAF [230], some of which are actively adopted for the evaluation of video algorithms, application products, and so on. On the other hand, existing DNN-based video coding approaches can adaptively optimize the efficiency of a predefined loss function, such as MSE, SSIM, adversarial loss [156], and VGG feature-based semantic loss. However, none of these loss functions has shown clear advantages. A unified, differentiable, and HVS-driven metric is of great importance for the capacity of DNN-based video coding techniques to offer perceptually better QoE.

The exponential growth of Internet traffic, a majority of which involves videos and images, has been the driving force for the development of video compression systems. The availability of a vast amount of images through the Internet, meanwhile, has been critical for the renaissance of the field of machine learning. In this work, we show that recent progress in deep learning can, in return, improve video compression. These mutual positive feedback suggest that significant progress could be achieved in both fields when they are investigated together. Therefore, the

[11] http://www.image-net.org/

approaches presented in this work could be the stepping stones for improving the compression efficiency in Internet-scale video applications.

From a different perspective, most compressed videos will be ultimately consumed by human beings or interpreted by machines, for subsequent task decisions. This is a typical CV problem, that is, content understanding and decisions for consumption or task-oriented application (e.g., detection and classification). Existing approaches have performed these tasks by first decoding the video and then examining the tasks via learned or rule-based methods on decoded pixels. Such separate processing, that is, video decoding followed by CV tasks, is relied upon mainly because traditional pixel-prediction-based differential video compression methods break the spatiotemporal features that could be potentially helpful for vision tasks. In contrast, recent DNN-based video compression algorithms rely on feature extraction, activation, suppression, and aggregation for more compact representation. For these reasons, it is expected that the CV tasks can be fulfilled in the compressive domain without bit decoding and pixel reconstruction. Our earlier attempts have shown a very encouraging gain in the accuracy of classification and retrieval in compressive formats, without resorting to the traditional feature-based approaches using decoded pixels [231], [232]. Using powerful DNNs to unify video compression and CV techniques is an exciting new field. It is also worth noting that the ISO/IEC MPEG is now actively working on a new project called "Video Coding for Machine" (VCM),[12] with emphasis on exploring video compression solutions for both human perception and machine intelligence. ■

REFERENCES

[1] D. J. Brady *et al.*, "Multiscale gigapixel photography," *Nature*, vol. 486, no. 7403, pp. 386–389, Jun. 2012.

[2] M. Cheng *et al.*, "A dual camera system for high spatiotemporal resolution video acquisition," *IEEE Trans. Pattern Anal. Mach. Intell.*, Mar. 2020.

[3] F. Dufaux, P. Le Callet, R. Mantiuk, and M. Mrak, *High Dynamic Range Video: From Acquisition, to Display and Applications*. New York, NY, USA: Academic, 2016.

[4] M. Winken, D. Marpe, H. Schwarz, and T. Wiegand, "Bit-depth scalable video coding," in *Proc. IEEE Int. Conf. Image Process.*, vol. 1, Sep. 2007, pp. I-5–I-8.

[5] P. Tudor, "Mpeg-2 video compression," *Electron. & Commun. Eng. J.*, vol. 7, no. 6, pp. 257–264, 1995.

[6] B. G. Haskell, A. Puri, and A. N. Netravali, *Digital Video: An Introduction to MPEG-2*. New York, NY, USA: Springer, 1996.

[7] T. Sikora, "The MPEG-4 video standard verification model," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 7, no. 1, pp. 19–31, Feb. 1997.

[8] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 3, pp. 301–317, Mar. 2001.

[9] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[10] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[11] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (HEVC)," in *Integrated Circuits and Systems, Algorithms and Architectures*, vol. 39. New York, NY, USA: Springer, 2014, pp. 49–90.

[12] G. J. Sullivan, P. N. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," *Proc. SPIE*, vol. 5558, pp. 454–474, Nov. 2004.

[13] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the stereo and multiview video coding extensions of the H.264/MPEG-4 AVC standard," *Proc. IEEE*, vol. 99, no. 4, pp. 626–642, Jan. 2011.

[14] L. Yu, S. Chen, and J. Wang, "Overview of AVS-video coding standards," *Signal Process., Image Commun.*, vol. 24, no. 4, pp. 247–262, Apr. 2009.

[15] S. Ma, S. Wang, and W. Gao, "Overview of IEEE 1857 video coding standard," in *Proc. IEEE Int. Conf. Image Process.*, Sep. 2013, pp. 1500–1504.

[16] J. Zhang, C. Jia, M. Lei, S. Wang, S. Ma, and W. Gao, "Recent development of AVS video coding standard: AVS3," in *Proc. Picture Coding Symp. (PCS)*, Nov. 2019, pp. 1–5.

[17] Y. Chen *et al.*, "An overview of core coding tools in the AV1 video codec," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 41–45.

[18] J. Han *et al.*, "A technical overview of AV1," 2020, *arXiv:2008.06091*. [Online]. Available: http://arxiv.org/abs/2008.06091

[19] *AOM—Alliance for Open Media*. Accessed: Sep. 5, 2020. [Online]. Available: http://www.aomedia.org/

[20] A. Aaron, Z. Li, M. Manohara, J. De Cock, and D. Ronca. Per-Title Encode Optimization. The Netflix Tech Blog. Dec. 14, 2015. [Online]. Available: https://netflixtechblog.com/per-title-encode-optimization-7e99442b62a2

[21] T. Shoham, D. Gill, S. Carmel, N. Terterov, and P. Tiktov, "Content-adaptive frame level rate control for video encoding using a perceptual video quality measure," in *Proc. SPIE, Appl. Digit. Image Process. XLII*, vol. 11137, Sep. 2019, Art. no. 111370Q.

[22] Y.-C. Lin, H. Denman, and A. Kokaram, "Multipass encoding for reducing pulsing artifacts in cloud based video transcoding," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2015, pp. 907–911.

[23] G. J. Sullivan and T. Wiegand, "Video compression-from concepts to the H.264/AVC standard," *Proc. IEEE*, vol. 93, no. 1, pp. 18–31, 2005.

[24] A. Norkin *et al.*, "HEVC deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1746–1754, Dec. 2012.

[25] C.-M. Fu *et al.*, "Sample adaptive offset in the HEVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1755–1764, Dec. 2012.

[26] R. Gupta, M. T. Khanna, and S. Chaudhury, "Visual saliency guided video compression algorithm," *Signal Process., Image Commun.*, vol. 28, no. 9, pp. 1006–1022, Oct. 2013.

[27] S. Liu *et al.*, *AHG on Neural Network Based Coding Tools*, document JVET-S0267/M54764, Joint Video Expert Team, Jun. 2020.

[28] S. Liu, E. Alshina, J. Pfaff, M. Wien, P. Wu, and Y. Ye, *Report of AHG11 Meeting on Neural Network-Based Video Coding*, document JVET-T0042/M54848, Joint Video Expert Team, Jul. 2020.

[29] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, Jul. 2017.

[30] C. Tian, Y. Xu, L. Fei, and K. Yan, "Deep learning for image denoising: A survey," in *Proc. Int. Conf. Genetic Evol. Comput.* New York, NY, USA: Springer, 2018, pp. 563–572.

[31] A. Chakrabarti, "A neural approach to blind motion deblurring," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 221–235.

[32] J. Koh, J. Lee, and S. Yoon, "Single-image deblurring with neural networks: A comparative survey," *Comput. Vis. Image Understand.*, vol. 203, Feb. 2021, Art. no. 103134.

[33] Y. Zhu, X. Fu, and A. Liu, "Learning dual transformation networks for image contrast enhancement," *IEEE Signal Process. Lett.*, vol. 27, pp. 1999–2003, 2020.

[34] W. Guan, T. Wang, J. Qi, L. Zhang, and H. Lu, "Edge-aware convolution neural network based salient object detection," *IEEE Signal Process. Lett.*, vol. 26, no. 1, pp. 114–118, Jan. 2019.

[35] L. Xu, J. Ren, Q. Yan, R. Liao, and J. Jia, "Deep edge-aware filters," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1669–1678.

[36] L. Zhaoping, "A new framework for understanding vision from the perspective of the primary visual cortex," *Current Opinion Neurobiol.*, vol. 58, pp. 1–10, Oct. 2019.

[37] X. Chen, M. Zirnsak, G. M. Vega, E. Govil, S. G. Lomber, and T. Moore, "Parietal cortex regulates visual salience and salience-driven behavior," *Neuron*, vol. 106, no. 1, pp. 187.e4–187.e4, Apr. 2020, doi: 10.1016/j.neuron.2020.01.016.

[38] O. Schwartz and E. Simoncelli, "Natural signal statistics and sensory gain control," *Nature Neurosci.*, vol. 4, no. 8, p. 819, 2001.

[39] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, Nov. 1998.

[40] L. Itti, "Automatic foveation for video compression using a neurobiological model of visual attention," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1304–1318, Oct. 2004.

[41] T. V. Nguyen, M. Xu, G. Gao, M. Kankanhalli, Q. Tian, and S. Yan, "Static saliency vs. dynamic saliency: A comparative study," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 987–996.

[42] E. Vig, M. Dorr, and D. Cox, "Large-scale optimization of hierarchical features for saliency prediction in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2798–2805.

[43] N. Liu, J. Han, D. Zhang, S. Wen, and T. Liu, "Predicting eye fixations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 362–370.

[12] https://mpeg.chiariglione.org/standards/exploration/video-coding-machines

[44] G. Li and Y. Yu, "Deep contrast learning for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 478–487.

[45] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 678–686.

[46] Q. Hou, M.-M. Cheng, X. Hu, A. Borji, Z. Tu, and P. Torr, "Deeply supervised salient object detection with short connections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3203–3212.

[47] B. Yan, H. Wang, X. Wang, and Y. Zhang, "An accurate saliency prediction method based on generative adversarial networks," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 2339–2343.

[48] Y. Xu, S. Gao, J. Wu, N. Li, and J. Yu, "Personalized saliency and its prediction," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 12, pp. 2975–2989, Dec. 2019.

[49] L. Bazzani, H. Larochelle, and L. Torresani, "Recurrent mixture density network for spatiotemporal visual attention," 2016, *arXiv:1603.08199*. [Online]. Available: http://arxiv.org/abs/1603.08199

[50] C. Bak, A. Kocak, E. Erdem, and A. Erdem, "Spatio-temporal saliency networks for dynamic saliency prediction," *IEEE Trans. Multimedia*, vol. 20, no. 7, pp. 1688–1698, Jul. 2018.

[51] M. Sun, Z. Zhou, Q. Hu, Z. Wang, and J. Jiang, "SG-FCN: A motion and memory-based deep learning model for video saliency detection," *IEEE Trans. Cybern.*, vol. 49, no. 8, pp. 2900–2911, Aug. 2019.

[52] L. Jiang, M. Xu, T. Liu, M. Qiao, and Z. Wang, "Deepvs: A deep learning based video saliency prediction approach," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 602–617.

[53] Z. Wang, J. Ren, D. Zhang, M. Sun, and J. Jiang, "A deep-learning based feature hybrid framework for spatiotemporal saliency detection inside videos," *Neurocomputing*, vol. 287, pp. 68–83, Apr. 2018.

[54] R. Cong, J. Lei, H. Fu, F. Porikli, Q. Huang, and C. Hou, "Video saliency detection via sparsity-based reconstruction and propagation," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4819–4831, Oct. 2019.

[55] K. Min and J. Corso, "TASED-net: Temporally-aggregating spatial encoder-decoder network for video saliency detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 2394–2403.

[56] W. Wang, J. Shen, J. Xie, M.-M. Cheng, H. Ling, and A. Borji, "Revisiting video saliency prediction in the deep learning era," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 1, pp. 220–237, Jan. 2021.

[57] A. Vetro, T. Haga, K. Sumi, and H. Sun, "Object-based coding for long-term archive of surveillance video," in *Proc. Int. Conf. Multimedia Expo*, Baltimore, MD, USA, vol. 2, Jul. 2003, p. 417.

[58] T. Nishi and H. Fujiyoshi, "Object-based video coding using pixel state analysis," in *Proc. 17th Int. Conf. Pattern Recognit.*, Cambridge, U.K., vol. 3, Aug. 2004, pp. 306–309.

[59] L. Zhu and Q. Zhang, "Motion-based foreground extraction in compressed video," in *Proc. Int. Conf. Measuring Technol. Mechatronics Autom.*, Mar. 2010, pp. 711–714.

[60] Z. Zhang, T. Jing, J. Han, Y. Xu, and X. Li, "Flow-process foreground region of interest detection method for video codecs," *IEEE Access*, vol. 5, pp. 16263–16276, 2017.

[61] Y. Guo, Z. Xuan, and L. Song, "Foreground target extraction method based on neighbourhood pixel intensity correction," *Austral. J. Mech. Eng.*, pp. 1–10, Apr. 2019.

[62] A. Shahbaz, V.-T. Hoang, and K.-H. Jo, "Convolutional neural network based foreground segmentation for video surveillance systems," in *Proc. IECON 45th Annu. Conf. IEEE Ind. Electron. Soc.*, vol. 1, Oct. 2019, pp. 86–89.

[63] S. Zhou, J. Wang, D. Meng, Y. Liang, Y. Gong, and N. Zheng, "Discriminative feature learning with foreground attention for person re-identification," *IEEE Trans. Image Process.*, vol. 28, no. 9, pp. 4671–4684, Sep. 2019.

[64] M. Babaee, D. T. Dinh, and G. Rigoll, "A deep convolutional neural network for background subtraction," 2017, *arXiv:1702.01731*. [Online]. Available: http://arxiv.org/abs/1702.01731

[65] X. Liang, S. Liao, X. Wang, W. Liu, Y. Chen, and S. Z. Li, "Deep background subtraction with guided learning," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.

[66] S. Zhang, K. Wei, H. Jia, X. Xie, and W. Gao, "An efficient foreground-based surveillance video coding scheme in low bit-rate compression," in *Proc. Vis. Commun. Image Process.*, Nov. 2012, pp. 1–6.

[67] H. Hadizadeh and I. V. Bajić, "Saliency-aware video compression," *IEEE Trans. Image Process.*, vol. 23, no. 1, pp. 19–33, Jan. 2014.

[68] Y. Li, W. Liao, J. Huang, D. He, and Z. Chen, "Saliency based perceptual HEVC," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2014, pp. 1–5.

[69] C. Ku, G. Xiang, F. Qi, W. Yan, Y. Li, and X. Xie, "Bit allocation based on visual saliency in HEVC," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2019, pp. 1–4.

[70] S. Zhu and Z. Xu, "Spatiotemporal visual saliency guided perceptual high efficiency video coding with neural network," *Neurocomputing*, vol. 275, pp. 511–522, Jan. 2018.

[71] V. Lyudvichenko, M. Erofeev, A. Ploshkin, and D. Vatolin, "Improving video compression with deep visual-attention models," in *Proc. Int. Conf. Intell. Med. Image Process. (IMIP)*, 2019, pp. 88–94.

[72] M. Cornia, L. Baraldi, G. Serra, and R. Cucchiara, "Predicting human eye fixations via an LSTM-based saliency attentive model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 5142–5154, Oct. 2018.

[73] X. Sun, X. Yang, S. Wang, and M. Liu, "Content-aware rate control scheme for HEVC based on static and dynamic saliency detection," *Neurocomputing*, vol. 411, pp. 393–405, Oct. 2020.

[74] M. Carandini, "Do we know what the early visual system does?" *J. Neurosci.*, vol. 25, no. 46, pp. 10577–10597, Nov. 2005.

[75] J. Kremkow *et al.*, "Neuronal nonlinearity explains greater visual spatial resolution for darks than lights," *Proc. Nat. Acad. Sci. USA*, vol. 111, no. 8, pp. 3170–3175, Feb. 2014.

[76] J. Ukita, T. Yoshida, and K. Ohki, "Characterisation of nonlinear receptive fields of visual neurons by convolutional neural network," *Sci. Rep.*, vol. 9, no. 1, pp. 1–17, Dec. 2019.

[77] P. Neri, "Nonlinear characterization of a simple process in human vision," *J. Vis.*, vol. 9, no. 12, p. 1, Nov. 2009.

[78] D. J. Heeger, "Normalization of cell responses in cat striate cortex," *Vis. Neurosci.*, vol. 9, no. 2, pp. 181–197, Aug. 1992.

[79] N. J. Priebe and D. Ferster, "Mechanisms of neuronal computation in mammalian visual cortex," *Neuron*, vol. 75, no. 2, pp. 194–208, Jul. 2012.

[80] M. Carandini and D. J. Heeger, "Normalization as a canonical neural computation," *Nature Rev. Neurosci.*, vol. 13, no. 1, p. 51, 2012.

[81] M. H. Turner and F. Rieke, "Synaptic rectification controls nonlinear spatial integration of natural visual inputs," *Neuron*, vol. 90, no. 6, pp. 1257–1271, Jun. 2016.

[82] D. Doshkov and P. Ndjiki-Nya, "Chapter 6—How to use texture analysis and synthesis methods for video compression," in *Academic Press Library in Signal Processing* (Academic Press Library in Signal Processing), vol. 5, S. Theodoridis and R. Chellappa, Eds. Oxford, U.K.: Elsevier, 2014, pp. 197–225.

[83] P. Ndjiki-Nya, D. Doshkov, H. Kaprykowsky, F. Zhang, D. Bull, and T. Wiegand, "Perception-oriented video coding based on image analysis and completion: A review," *Signal Process., Image Commun.*, vol. 27, no. 6, pp. 579–594, Jul. 2012.

[84] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using Gabor filters," in *Proc. Int. Conf. Syst., Man, Cybern. Conf.*, Los Angeles, CA, USA, 1990, pp. 14–19.

[85] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 12, no. 1, pp. 55–73, Jan. 1990.

[86] U. S. Thakur and O. Chubach, "Texture analysis and synthesis using steerable pyramid decomposition for video coding," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, London, U.K., Sep. 2015, pp. 204–207.

[87] J. Portilla and E. P. Simoncelli, "A parametric texture model based on joint statistics of complex wavelet coefficients," *Int. J. Comput. Vis.*, vol. 40, no. 1, pp. 49–70, 2000.

[88] S. Bansal, S. Chaudhury, and B. Lall, "Dynamic texture synthesis for video compression," in *Proc. Nat. Conf. Commun. (NCC)*, New Delhi, India, Feb. 2013, pp. 1–5.

[89] G. R. Cross and A. K. Jain, "Markov random field texture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-5, no. 1, pp. 25–39, Jan. 1983.

[90] R. Chellappa and S. Chatterjee, "Classification of textures using Gaussian Markov random fields," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 33, no. 4, pp. 959–963, Aug. 1985.

[91] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[92] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded up robust features," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2006, pp. 404–417.

[93] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[94] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[95] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3828–3836.

[96] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2010, pp. 143–156.

[97] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Kerkyra, Greece, vol. 2, Sep. 1999, pp. 1033–1038.

[98] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proc. 27th Annu. Conf. Comput. Graph. Interact. Techn.*, New Orleans, LA, USA, 2000, pp. 479–488.

[99] M. Ashikhmin, "Synthesizing natural textures," in *Proc. Symp. Interact. 3D Graph.*, New York, NY, USA, 2001, pp. 217–226.

[100] H. Derin and H. Elliott, "Modeling and segmentation of noisy and textured images using Gibbs random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 1, pp. 39–55, Jan. 1987.

[101] D. J. Heeger and J. R. Bergen, "Pyramid-based texture analysis/synthesis," in *Proc. 22nd Annu. Conf. Comput. Graph. Interact. Techn.*, Los Angeles, CA, USA, 1995, pp. 229–238.

[102] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 262–270.

[103] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 702–716.

[104] T. Chen, H. Liu, Q. Shen, T. Yue, X. Cao, and Z. Ma, "DeepCoder: A deep neural network based video compression," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, St. Petersburg, FL, USA, Dec. 2017, pp. 1–4.

[105] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," 2016, *arXiv:1611.01704*, [Online]. Available: https://arxiv.org/abs/1611.01704

[106] D. Liu, Y. Li, J. Lin, H. Li, and F. Wu, "Deep learning-based video coding: A review and a case study," *ACM Comput. Surv.*, vol. 53, no. 1, pp. 1–35, 2020.

[107] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, "Image and video compression with neural networks: A review," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 6, pp. 1683–1698, Jun. 2020.

[108] Y. Li *et al.*, "Convolutional neural network-based block up-sampling for intra frame coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 9, pp. 2316–2330, Sep. 2018.

[109] F. Jiang, W. Tao, S. Liu, J. Ren, X. Guo, and D. Zhao, "An end-to-end compression framework based on convolutional neural networks," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 3007–3018, Oct. 2018.

[110] M. Afonso, F. Zhang, and D. R. Bull, "Video compression based on spatio-temporal resolution adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 275–280, Jan. 2019.

[111] J. Lin, D. Liu, H. Yang, H. Li, and F. Wu, "Convolutional neural network-based block up-sampling for HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 12, pp. 3701–3715, Dec. 2019.

[112] W. Yang, X. Zhang, Y. Tian, W. Wang, J.-H. Xue, and Q. Liao, "Deep learning for single image super-resolution: A brief review," *IEEE Trans. Multimedia*, vol. 21, no. 12, pp. 3106–3121, Dec. 2019.

[113] W. Cui, T. Zhang, S. Zhang, F. Jiang, W. Zuo, and D. Zhao, "Convolutional neural networks based intra prediction for HEVC," 2018, *arXiv:1808.05734*. [Online]. Available: http://arxiv.org/abs/1808.05734

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

[115] J. Pfaff *et al.*, "Neural network based intra prediction for video coding," *Proc. SPIE*, vol. 10752, Sep. 2018, Art. no. 1075213.

[116] Y. Hu, W. Yang, M. Li, and J. Liu, "Progressive spatial recurrent neural network for intra prediction," *IEEE Trans. Multimedia*, vol. 21, no. 12, pp. 3024–3037, Dec. 2019.

[117] Z. Jin, P. An, and L. Shen, "Video intra prediction using convolutional encoder decoder network," *Neurocomputing*, vol. 394, pp. 168–177, Jun. 2020.

[118] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," *IEEE Trans. Commun.*, vol. 41, no. 4, pp. 604–612, Apr. 1993.

[119] N. Yan, D. Liu, H. Li, and F. Wu, "A convolutional neural network approach for half-pel interpolation in video coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Baltimore, MD, USA, May 2017, pp. 1–4.

[120] H. Zhang, L. Song, Z. Luo, and X. Yang, "Learning a convolutional neural network for fractional interpolation in HEVC inter coding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, St. Petersburg, FL, USA, Dec. 2017, pp. 1–4.

[121] J. Liu, S. Xia, W. Yang, M. Li, and D. Liu, "One-for-all: Grouped variation network-based fractional interpolation in video coding," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2140–2151, May 2019.

[122] L. Zhao, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Enhanced motion-compensated video coding with deep virtual reference frame generation," *IEEE Trans. Image Process.*, vol. 28, no. 10, pp. 4832–4844, Oct. 2019.

[123] S. Xia, W. Yang, Y. Hu, and J. Liu, "Deep inter prediction via pixel-wise motion oriented reference generation," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 1710–1774.

[124] S. Huo, D. Liu, F. Wu, and H. Li, "Convolutional neural network-based motion compensation refinement for video coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Florence, Italy, 2018, pp. 1–4.

[125] M. M. Alam, T. D. Nguyen, M. T. Hagan, and D. M. Chandler, "A perceptual quantization strategy for HEVC based on a convolutional neural network trained on natural images," *Proc. SPIE*, vol. 9599, Sep. 2015, Art. no. 959918.

[126] R. Song, D. Liu, H. Li, and F. Wu, "Neural network-based arithmetic coding of intra prediction modes in HEVC," in *Proc. IEEE Vis. Commun. Image Process.*, St. Petersburg, FL, USA, Dec. 2017, pp. 1–4.

[127] S. Puri, S. Lasserre, and P. Le Callet, "CNN-based transform index prediction in multiple transforms framework to assist entropy coding," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Kos Island, Greece, Aug. 2017, pp. 798–802.

[128] C. Ma, D. Liu, X. Peng, and F. Wu, "Convolutional neural network-based arithmetic coding of DC coefficients for HEVC intra coding," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 1772–1776.

[129] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An end-to-end deep video compression framework," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Long Beach, CA, USA, Jun. 2019, pp. 11006–11015.

[130] O. Rippel, S. Nair, C. Lew, S. Branson, A. Anderson, and L. Bourdev, "Learned video compression," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3454–3463.

[131] H. Liu, H. Shen, L. Huang, M. Lu, T. Chen, and Z. Ma, "Learned video compression via joint spatial-temporal correlation exploration," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, hboxpp. 11580–11587.

[132] G. Toderici *et al.*, "Variable rate image compression with recurrent neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–12.

[133] G. Toderici *et al.*, "Full resolution image compression with recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Honolulu, HI, USA, Jul. 2017, pp. 5306–5314.

[134] N. Johnston *et al.*, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4385–4393.

[135] Y. Choi, M. El-Khamy, and J. Lee, "Variable rate deep image compression with a conditional autoencoder," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3146–3154.

[136] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, "End-to-end learnt image compression via non-local attention optimization and improved context modeling," *IEEE Trans. Image Process.*, Feb. 2021.

[137] J. Ballé, "Efficient nonlinear transforms for lossy image compression," in *Proc. IEEE Picture Coding Symp. (PCS)*, 2018, pp. 248–252.

[138] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," 2018, *arXiv:1809.10452*. [Online]. Available: https://arxiv.org/abs/1809.10452

[139] J. Klopp, Y.-C. F. Wang, S.-Y. Chien, and L.-G. Chen, "Learning a code-space predictor by exploiting intra-image-dependencies," in *Proc. BMVC*, 2018, p. 124.

[140] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool, "Conditional probability models for deep image compression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 4394–4402.

[141] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, "DVC: An end-to-end deep video compression framework," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2019, pp. 11006–11015.

[142] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–23.

[143] C.-Y. Wu, N. Singhal, and P. Krähenbühl, "Video compression through image interpolation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 416–431.

[144] A. Djelouah, J. Campos, S. Schaub-Meyer, and C. Schroers, "Neural inter-frame compression for video coding," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6421–6429.

[145] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," 2017, *arXiv:1703.10553*. [Online]. Available: http://arxiv.org/abs/1703.10553

[146] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.

[147] Y. Hu, W. Yang, and J. Liu, "Coarse-to-fine hyper-prior modeling for learned image compression," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11013–11020.

[148] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10794–10803.

[149] A. van den Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," 2016, *arXiv:1601.06759*. [Online]. Available: http://arxiv.org/abs/1601.06759

[150] S. Reed *et al.*, "Parallel multiscale autoregressive density estimation," in *Proc. 34th Int. Conf. Mach. Learn. (JMLR)*, 2017, pp. 2912–2921.

[151] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 7939–7948.

[152] J. Lee, S. Cho, and M. Kim, "An end-to-end joint learning scheme of image compression and quality enhancement with improved entropy minimization," 2019, *arXiv:1912.12817*. [Online]. Available: http://arxiv.org/abs/1912.12817

[153] O. Rippel and L. Bourdev, "Real-time adaptive image compression," 2017, *arXiv:1705.05823*. [Online]. Available: http://arxiv.org/abs/1705.05823

[154] C. Huang, H. Liu, T. Chen, Q. Shen, and Z. Ma, "Extreme image coding via multiscale autoencoders with generative adversarial optimization," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2019, pp. 1–4.

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

[156] H. Liu, T. Chen, Q. Shen, T. Yue, and Z. Ma, "Deep image compression via end-to-end learning," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops*, Jun. 2018, pp. 2575–2578.

[157] J. L. Hennessy and D. A. Patterson, "A new golden age for computer architecture," *Commun. ACM*, vol. 62, no. 2, pp. 48–60, Jan. 2019.

[158] S. Midtskogen and J.-M. Valin, "The Av1 constrained directional enhancement filter (CDEF)," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2018, pp. 1193–1197.

[159] D. Mukherjee, S. Li, Y. Chen, A. Anis, S. Parker, and J. Bankoski, "A switchable loop-restoration with side-information framework for the emerging AV1 video codec," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 265–269.

[160] C.-Y. Tsai *et al.*, "Adaptive loop filtering for video coding," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 934–945, Dec. 2013.

[161] W.-S. Park and M. Kim, "CNN-based in-loop filtering for coding efficiency improvement," in *Proc. IEEE 12th Image, Video, Multidimensional Signal Process. Workshop (IVMSP)*, Jul. 2016, pp. 1–5.

[162] Y. Dai, D. Liu, and F. Wu, "A convolutional neural

network approach for post-processing in HEVC intra coding," in *Proc. Int. Conf. Multimedia Model.* New York, NY, USA: Springer, 2017, pp. 28–39.
[163] Y. Zhang, T. Shen, X. Ji, Y. Zhang, R. Xiong, and Q. Dai, "Residual highway convolutional neural networks for in-loop filtering in HEVC," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3827–3841, Aug. 2018.
[164] X. Xu *et al.*, "Dense inception attention neural network for in-loop filter," in *Proc. Picture Coding Symp. (PCS)*, Nov. 2019, pp. 1–5.
[165] K. Lin *et al.*, "Residual in residual based convolutional neural network in-loop filter for AVS3," in *Proc. Picture Coding Symp. (PCS)*, Nov. 2019, pp. 1–5.
[166] J. Kang, S. Kim, and K. M. Lee, "Multi-modal/multi-scale convolutional neural network based in-loop filter design for next generation video codec," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 26–30.
[167] C. Jia, S. Wang, X. Zhang, S. Wang, and S. Ma, "Spatial-temporal residue network based in-loop filter for video coding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.
[168] X. Meng, C. Chen, S. Zhu, and B. Zeng, "A new HEVC in-loop filter based on multi-channel long-short-term dependency residual networks," in *Proc. Data Compress. Conf.*, Mar. 2018, pp. 187–196.
[169] D. Li and L. Yu, "An in-loop filter based on low-complexity CNN using residuals in intra video coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2019, pp. 1–5.
[170] D. Ding, L. Kong, G. Chen, Z. Liu, and Y. Fang, "A switchable deep learning approach for in-loop filtering in video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 1871–1887, Jul. 2020.
[171] D. Ding, G. Chen, D. Mukherjee, U. Joshi, and Y. Chen, "A CNN-based in-loop filtering approach for AV1 video codec," in *Proc. Picture Coding Symp. (PCS)*, Nov. 2019, pp. 1–5.
[172] G. Chen, D. Ding, D. Mukherjee, U. Joshi, and Y. Chen, "AV1 in-loop filtering using a wide-activation structured residual network," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Taipei, Taiwan, Sep. 2019, pp. 1725–1729.
[173] H. Yin, R. Yang, X. Fang, and S. Ma, *Ce13-1.2: Adaptive Convolutional Neural Network Loop Filter*, document JVET-N0480, 2019.
[174] T. Li, M. Xu, C. Zhu, R. Yang, Z. Wang, and Z. Guan, "A deep learning approach for multi-frame in-loop filter of HEVC," *IEEE Trans. Image Process.*, vol. 28, no. 11, pp. 5663–5678, Nov. 2019.
[175] C. Jia *et al.*, "Content-aware convolutional neural network for in-loop filtering in high efficiency video coding," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3343–3356, Jul. 2019.
[176] C. Dong, Y. Deng, C. C. Loy, and X. Tang, "Compression artifacts reduction by a deep convolutional network," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 576–584.
[177] L. Cavigelli, P. Hager, and L. Benini, "CAS-CNN: A deep convolutional neural network for image compression artifact suppression," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Anchorage, AK, USA, May 2017, pp. 752–759.
[178] J. Guo and H. Chao, "Building dual-domain representations for compression artifacts reduction," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 628–644.
[179] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Deep generative adversarial compression artifact removal," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 4826–4835.
[180] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, "Multi-level wavelet-CNN for image restoration," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Salt Lake City, UT, USA, Jun. 2018, pp. 773–782.
[181] Y. Zhang, L. Sun, C. Yan, X. Ji, and Q. Dai, "Adaptive residual networks for high-quality image restoration," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3150–3163, Jul. 2018.
[182] T. Wang, M. Chen, and H. Chao, "A novel deep learning-based method of improving coding efficiency from the decoder-end for HEVC," in *Proc. Data Compress. Conf. (DCC)*, Snowbird, UT, USA, Apr. 2017, pp. 410–419.
[183] R. Yang, M. Xu, and Z. Wang, "Decoder-side HEVC quality enhancement with scalable convolutional neural network," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 817–822.
[184] X. He, Q. Hu, X. Zhang, C. Zhang, W. Lin, and X. Han, "Enhancing HEVC compressed videos with a partition-masked convolutional neural network," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 216–220.
[185] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.
[186] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "FlowNet 2.0: Evolution of optical flow estimation with deep networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2462–2470.
[187] D. Sun, X. Yang, M.-Y. Liu, and J. Kautz, "PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 8934–8943.
[188] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *Int. J. Comput. Vis.*, vol. 127, no. 8, pp. 1106–1125, Aug. 2019.
[189] W. Bao, W.-S. Lai, X. Zhang, Z. Gao, and M.-H. Yang, "MEMC-net: Motion estimation and motion compensation driven neural network for video interpolation and enhancement," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 933–948, Mar. 2021.
[190] X. Wang, K. C. K. Chan, K. Yu, C. Dong, and C. C. Loy, "EDVR: Video restoration with enhanced deformable convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2019, pp. 1954–1963.
[191] R. Yang, M. Xu, Z. Wang, and T. Li, "Multi-frame quality enhancement for compressed video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6664–6673.
[192] Z. Guan, Q. Xing, M. Xu, R. Yang, T. Liu, and Z. Wang, "MFQE 2.0: A new approach for multi-frame quality enhancement on compressed video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 3, pp. 949–963, Mar. 2021.
[193] J. Tong, X. Wu, D. Ding, Z. Zhu, and Z. Liu, "Learning-based multi-frame video quality enhancement," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 929–933.
[194] M. Lu, M. Cheng, Y. Xu, S. Pu, Q. Shen, and Z. Ma, "Learned quality enhancement via multi-frame priors for HEVC compliant low-delay applications," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 934–938.
[195] U. Joshi *et al.*, "Novel inter and intra prediction tools under consideration for the emerging AV1 video codec," *Proc. SPIE*, vol. 10396, Sep. 2017, Art. no. 103960F.
[196] Z. Liu, D. Mukherjee, W.-T. Lin, P. Wilkins, J. Han, and Y. Xu, "Adaptive multi-reference prediction using a symmetric framework," *Electron. Imag.*, vol. 2017, no. 2, pp. 65–72, Jan. 2017.
[197] Y. Chen *et al.*, "An overview of core coding tools in the AV1 video codec," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 41–45.
[198] Y. Wang, S. Inguva, and B. Adsumilli, "YouTube UGC dataset for video compression research," in *Proc. IEEE 21st Int. Workshop Multimedia Signal Process. (MMSP)*, Kuala Lumpur, Malaysia, Sep. 2019, pp. 1–5.
[199] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2881–2890.
[200] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[201] M. Bosch, F. Zhu, and E. J. Delp, "Segmentation-based video compression using texture and motion models," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 7, pp. 1366–1377, Nov. 2011.
[202] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[203] T. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. IEEE Eur. Conf. Comput. Vis.*, Zürich, Switzerland, Sep. 2014, pp. 740–755.
[204] B. Zhou *et al.*, "Semantic understanding of scenes through the ADE20K dataset," *Int. J. Comput. Vis.*, vol. 127, no. 3, pp. 302–321, Mar. 2019.
[205] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.
[206] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," 2014, *arXiv:1412.7062*. [Online]. Available: http://arxiv.org/abs/1412.7062
[207] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," 2015, *arXiv:1511.07122*. [Online]. Available: http://arxiv.org/abs/1511.07122
[208] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba, "Scene parsing through ADE20K dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 633–641.
[209] D. Chen, Q. Chen, and F. Zhu, "Pixel-level texture segmentation based AV1 video compression," in *Proc. ICASSP—IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1622–1626.
[210] M. Bosch, F. Zhu, and E. J. Delp, "Spatial texture models for video compression," in *Proc. IEEE Int. Conf. Image Process.*, San Antonio, TX, USA, vol. 1, Sep. 2007, pp. 93–96.
[211] C. Fu, D. Chen, E. Delp, Z. Liu, and F. Zhu, "Texture segmentation based video compression using convolutional neural networks," *Electron. Imag.*, vol. 2018, no. 2, pp. 1–155, 2018.
[212] *Methodologies for the Subjective Assessment of the Quality of Television Images*, document BT.500-14, ITU-R Recommendation Geneva, Switzerland, 2019.
[213] M. Haindl and S. Mikes, "Texture segmentation benchmark," in *Proc. 19th Int. Conf. Pattern Recognit.*, Dec. 2008, pp. 1–4.
[214] N. Xu *et al.*, "YouTube-VOS: A large-scale video object segmentation benchmark," 2018, *arXiv:1809.03327*. [Online]. Available: http://arxiv.org/abs/1809.03327
[215] F. Perazzi, J. Pont-Tuset, B. McWilliams, L. Van Gool, M. Gross, and A. Sorkine-Hornung, "A benchmark dataset and evaluation methodology for video object segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 724–732.
[216] H. Liu *et al.*, "Neural video coding using multiscale motion compensation and spatiotemporal context model," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Nov. 3, 2020, doi: 10.1109/TCSVT.2020.3035680.
[217] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3214–3223.
[218] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 2012.
[219] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu, "Residual non-local attention networks for image restoration," 2019, *arXiv:1903.10082*. [Online]. Available: http://arxiv.org/abs/1903.10082
[220] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process.*

Syst., vol. 28, 2015, pp. 802–810.
[221] G. Lu, X. Zhang, W. Ouyang, L. Chen, Z. Gao, and D. Xu, "An end-to-end learning framework for video compression," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 20, 2020, doi: 10.1109/TPAMI.2020.2988453.
[222] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang, "Ntire 2017 challenge on single image super-resolution: Methods and results," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jul. 2017, pp. 114–125.
[223] J. Yu *et al.*, "Wide activation for efficient and accurate image super-resolution," 2018, *arXiv:1808.08718*. [Online]. Available: http://arxiv.org/abs/1808.08718
[224] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* New York, NY, USA: Springer, 2016, pp. 630–645.
[225] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
[226] Q. Xia, H. Liu, and Z. Ma, "Object-based image coding: A learning-driven revisit," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.
[227] Y. Wang, Q. Yao, J. T. Kwok, and L. M. Ni, "Generalizing from a few examples: A survey on few-shot learning," *ACM Comput. Surv.*, vol. 53, no. 3, pp. 1–34, 2020.
[228] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *Proc. 37th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, USA, vol. 2, Nov. 2003, pp. 1398–1402.
[229] D. Yuan, T. Zhao, Y. Xu, H. Xue, and L. Lin, "Visual JND: A perceptual measurement in video coding," *IEEE Access*, vol. 7, pp. 29014–29022, 2019.
[230] Netflix, Inc. (2017). *VMAF: Perceptual Video Quality Assessment Based on Multi-Method Fusion*. [Online]. Available: https://github.com/Netflix/vmaf
[231] Q. Shen *et al.*, "*Codedvision*: Towards joint image understanding and compression via end-to-end learning," in *Proc. Pacific Rim Conf. Multimedia*. New York, NY, USA: Springer, 2018, pp. 3–14.
[232] L. Liu, H. Liu, T. Chen, Q. Shen, and Z. Ma, "Codedretrieval: Joint image compression and retrieval with neural networks," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2019, pp. 1–4.

ABOUT THE AUTHORS

Dandan Ding (Member, IEEE) received the B.Eng. degree (honors) in communication engineering and the Ph.D. degree from Zhejiang University, Hangzhou, China, in 2006 and June 2011, respectively.

From 2007 to 2008, she was an Exchange Student with the Microelectronic Systems Laboratory (GR-LSM), École Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland. She was a Postdoctoral Researcher with Zhejiang University from July 2011 to June 2013, where she was a Research Associate till 2015. Since 2016, she has been a Faculty Member with tenure track with the Department of Information Science and Engineering, Hangzhou Normal University, Hangzhou. Her research interests include artificial intelligence-based image/video processing, video coding algorithm design and optimization, and SoC design. Currently, she is active in the development of new video coding algorithms for the new generation video coding standards, such as AOM/AV2.

Dr. Ding has continuously received the annual research grant sponsored by Google's Chrome University Relationship Program (CURP) since 2018. She was invited to the First AOM Global Research Symposium in October 2019 to present her research work in AV1.

Zhan Ma (Senior Member, IEEE) received the B.S. and M.S. degrees from the Huazhong University of Science and Technology, Wuhan, China, in 2004 and 2006, respectively, and the Ph.D. degree from New York University, New York, NY, USA, in 2011.

From 2011 to 2014, he was with Samsung Research America, Dallas, TX, USA, and with Futurewei Technologies, Inc., Santa Clara, CA, USA. He is currently the faculty of the Electronic Science and Engineering School, Nanjing University, Jiangsu, China. His research interests include learning-based image/video coding and computational imaging.

Dr. Ma was awarded the 2018 PCM Best Paper Finalist, the 2019 IEEE Broadcast Technology Society Best Paper Award, and the 2020 IEEE MMSP Grand Challenge Best Image Coding Solution.

Di Chen (Member, IEEE) received the B.S. degree from the Huazhong University of Science and Technology, Wuhan, China, in 2012, the M.S. degree from the Rensselaer Polytechnic Institute, Troy, NY, USA, in 2014, and the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2020.

She is currently a Software Engineer with the Media Algorithm Team, Google, Mountain View, CA, USA. Her research focuses on image and video processing, and video coding algorithm design and optimization.

Qingshuang Chen (Member, IEEE) received the B.S. degree in electrical engineering from Purdue University, West Lafayette, IN, USA, in 2014, where she is currently working toward the Ph.D. degree at the Video and Image Processing Laboratory (VIPER).

Her current research interests include learning-based image/video compression and video analysis using deep neural networks.

Zoe Liu received the B.E. degree (Honsors) and the M.E./Ph.D. degree from Tsinghua University, Beijing, China, in 1995 and 2000, respectively, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 2004, all in electrical engineering.

She was a Software Engineer with the Google WebM Team, Mountain View, CA, USA, and has been a key contributor to the newly finalized royalty-free video codec standard AOM/AV1. She has been devoted to the design and development of innovative products in the field of video codec and real-time video communications for almost 20 years. She is the Co-Founder and the President of Visionular Inc., Palo Alto, CA, USA, a startup delivering cutting-edge video solutions to enterprise customers worldwide. She was a 2018 Google I/O speaker. She has published approximately 50 international conference papers and journal articles. Her main research interests include video compression, image processing, and machine learning.

Fengqing Zhu (Senior Member, IEEE) received the B.S.E.E. (honors), M.S., and Ph.D. degrees in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 2004, 2006, and 2011, respectively.

Prior to joining Purdue University in 2015, she was a Staff Researcher with Futurewei Technologies, Inc., Santa Clara, CA, USA. She is currently an Assistant Professor of electrical and computer engineering with Purdue University. Her research interests include image processing and analysis, video compression, and computer vision.

Dr. Zhu received the Certification of Recognition for Core Technology Contribution from Futurewei Technologies in 2012. She was a recipient of the NSF CISE Research Initiation Initiative (CRII) Award in 2017 and the Google Faculty Research Award in 2019.