Towards Modality Transferable Visual Information Representation with Optimal Model Compression

Rongqun Lin
Department of Computer Science,
City University of Hong Kong
rqlin3-c@my.cityu.edu.hk

Linwei Zhu
Shenzhen Institutes of Advanced Technology,
Chinese Academy of Sciences
lw.zhu@siat.ac.cn

Shiqi Wang*
Department of Computer Science,
City University of Hong Kong
shiqwang@cityu.edu.hk

Sam Kwong*
Department of Computer Science,
City University of Hong Kong
cssamk@cityu.edu.hk

ABSTRACT

Compactly representing the visual signals is of fundamental importance in various image/video-centered applications. Although numerous approaches were developed for improving the image and video coding performance by removing the redundancies within visual signals, much less work has been dedicated to the transformation of the visual signals to another well-established modality for better representation capability. In this paper, we propose a new scheme for visual signal representation that leverages the philosophy of transferable modality. In particular, the deep learning model, which characterizes and absorbs the statistics of the input scene with online training, could be efficiently represented in the sense of rate-utility optimization to serve as the enhancement layer in the bitstream. As such, the overall performance can be further guaranteed by optimizing the new modality incorporated. The proposed framework is implemented on the state-of-the-art video coding standard (i.e., versatile video coding), and significantly better representation capability has been observed based on extensive evaluations.

CCS CONCEPTS

• **Computing methodologies → Image processing**.

KEYWORDS

Deep learning; visual signal representation; deep learning model communication; rate–utility optimization

ACM Reference Format:
Rongqun Lin, Linwei Zhu, Shiqi Wang, and Sam Kwong. 2020. Towards Modality Transferable Visual Information Representation with Optimal Model Compression. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3394171.3413762

1 INTRODUCTION

Recently, we have witnessed exponential growth of image/video services, coinciding with the accelerated proliferation of acquisition and display devices. The gigantic scale of visual data motivates the research of compact signal representation, which is the long-standing problem and indispensable in numerous applications. Traditional methods aim to remove the redundancies within the visual signals, such as spatial, temporal, statistical and perceptual redundancies. Based on the philosophy of redundancy removal, a series of video coding standards have been developed, including H.264/AVC [70], H.265/HEVC [61], VP9[50], AV1[14], H.266/VVC [11] and AVS [24].

With the surge of deep learning, numerous efforts have been devoted to improving the compact signal representation capability with deep neural networks, including incorporating the neural network into the hybrid video coding framework [22, 32, 34, 36, 37, 41–43, 49, 53, 54, 60, 73, 80–83] and end-to-end compression [3, 5–7, 57, 58, 64, 65]. In the first category, intra prediction, inter prediction, loop filtering and entropy coding modules have been significantly enhanced with the deep neural networks at both video encoder and decoder sides. In the second category, the visual information is compactly represented with the latent code in the manner of end-to-end training. Though promising performance has been achieved, the systematical study regarding the redundancy removal by transferring from visual information to deep learning model which is recognized as one important modality of knowledge [16] on data statistics has been largely ignored.

The deep neural networks have been regarded as the important modality of knowledge in Knowledge Centric Networking (KCN) [71], and the network communication has been widely studied in the literature [15]. In video delivery [75, 76], the network has been learned and coded in an online manner, which significantly improves the performance of video streaming. In this paper, based on the extensive studies on quality enhancement with neural networks, we make a further attempt by optimally transferring the visual signals to another well-established modality deep neural network for better signal representation capability. We aim to explore the possibility of efficient representation of the visual information with deep learning model in the sense of rate-utility optimization, such that

*Shiqi Wang and Sam Kwong are the corresponding authors.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413762

the model information could serve as an enhancement layer in the representation bitstream. As such, instead of internally removing the redundancies of the visual content, the visual information is further represented with the assistant of the knowledge in deep learning models. The contributions of this paper are summarized as follows.

- We propose to leverage the representation capability of deep neural networks for further visual redundancy removal on top of the state-of-the-art compression framework. The proposed scheme is designed based on the philosophy of online modality transfer with model compression and optimal model selection.
- We propose to efficiently compress the deep learning model with an effort to optimize the transferable modality. Instead of quantizing the weight after online model training, weight quantization has been incorporated by the scale transform and affine transform during the online training, such that the model representation is optimized in the training phase, leading to reproducible performance in the testing phase.
- We propose to optimize the model representation with rate-utility optimization. In particular, instead of only ensuring the optimal signal representation capability, the model rate is also considered in the optimization process. As such, the overall performance can be ensured by optimal model selection and representation.

2 RELATED WORKS

2.1 Image/video compression

Traditional compact visual information representation relies on image/video compression, and recently numerous image/video coding standards have been developed based on the hybrid coding framework, such as JPEG [66], H.264/AVC [70], H.265/HEVC [61], VP9 [50], AV1 [14], AVS [24] and H.266/VVC [11]. In these standards, the spatial, temporal, and statistical redundancies have been fully exploited to improve the coding performance.

Regarding spatial redundancy, the state-of-the-art intra coding extends the number of angular prediction modes to 67 [18] for better capturing the arbitrary texture directions. Moreover, Multiple Reference Line (MRL) [13] intra prediction is adopted where more informative reference lines are involved in the conventional intra prediction procedure, leading to further removal of the spatial redundancy. To remove temporal redundancy, affine motion compensated prediction [12] has been introduced to improve the precision of irregular motions, such as zoom in/out and rotation.

For the removal of the statistical redundancy, Context-Adaptive Binary Arithmetic Coding (CABAC) [63] is an efficient entropy coding method, which has been used in H.264/AVC, H.265/HEVC, AVS, VVC etc. CABAC combines the adaptive binary arithmetic coding with the context modeling, which brings sufficient adaptation and redundancy reduction in a lossless way. Additionally, Adaptive Loop Filter (ALF) [12] is placed at the last stage of the codec, which is a Wiener-filter targeting at minimizing the mean squared error between the original and reconstructed frames. The entire coding process is optimized with rate-distortion optimization [62] to ensure the optimal coding performance.

2.2 Neural network compression

Recently, numerous efforts have been devoted to neural network compression [17], aiming to lessen the massive cost of deep neural network in terms of both storage and computation without significant degradation on the performance. Typically, these approaches can be divided into eight categories: 1) parameters pruning and filter selection, 2) quantization, 3) matrix factorization, 4) transferred convolutional filters, 5) knowledge transfer and distillation, 6) network redesign, 7) transparent compression and 8) entropy constrains.

In particular, parameter pruning [23, 26, 27] aims to prune the unnecessary or weak response neurons, and filter selection approaches [28, 38, 68, 74] attempt to abolish unimportant channels. Weight quantization [19, 25, 33, 40, 44, 46, 52, 55, 56, 72, 84] quantizes the weights of neural network into binary, ternary values or their powers with little degradation on accuracy. Considering the whole neuron weight as a matrix, matrix factorization [45, 48, 78] can further reconstruct the weight matrix with the low rank methods. The design philosophy behind transferred convolutional filters based methods [59, 79] adopts special structural convolutional filters to shrink the parameters. Regarding the knowledge distilling [4, 29, 47], a small scale network can be learned under the guidance of a large scale teacher network. Furthermore, many new network architectures have been proposed by redesigning the network structure, such as BinaryNet [20], XNORNet [55], SqueezeNet [35] and MobileNet [30]. Transparent compression method [39] adopts the transform coding strategies without modifying the network structure. Methods based on entropy constrains on parameters [51, 69] utilize entropy penalized policy to produce compact representation of model.

2.3 Deep learning based image/video coding

With the surge of deep learning in many applications, numerous deep learning based image/video compression approaches have been proposed to achieve more compact representation of visual signals. The majority of deep learning based video coding approaches fall into two categories: end-to-end compression and the substitution of the modules with deep neural networks in the hybrid coding framework.

End-to-end compression technologies explore the representation capacities of deep neural networks which could be end-to-end trained. As such, a better trade-off between the bitrate and the distortion can be achieved. The pioneering work [64] utilizes Recurrent Neural Network (RNN) to reconstruct the image in an end-to-end manner. The Convolutional Neural Network (CNN) based end-to-end image compression methods [5–7] realize effective compression through Generalized Divisive Normalization (GDN) nonlinearity embedded analysis and synthesis transforms. The pioneering approach utilizing the adversarial loss function for image compression was proposed by Rippel *et al.* [57]. Subsequently, the Generative Adversarial Networks (GANs) have been applied to pursue realistic reconstruction quality of images with very low bitrates [3, 58, 65].

The deep neural networks can also be embedded into main modules in the hybrid coding framework to improve the coding performance, i.e., intra prediction, inter-prediction, entropy coding and loop filtering. Intra prediction techniques using deep neural

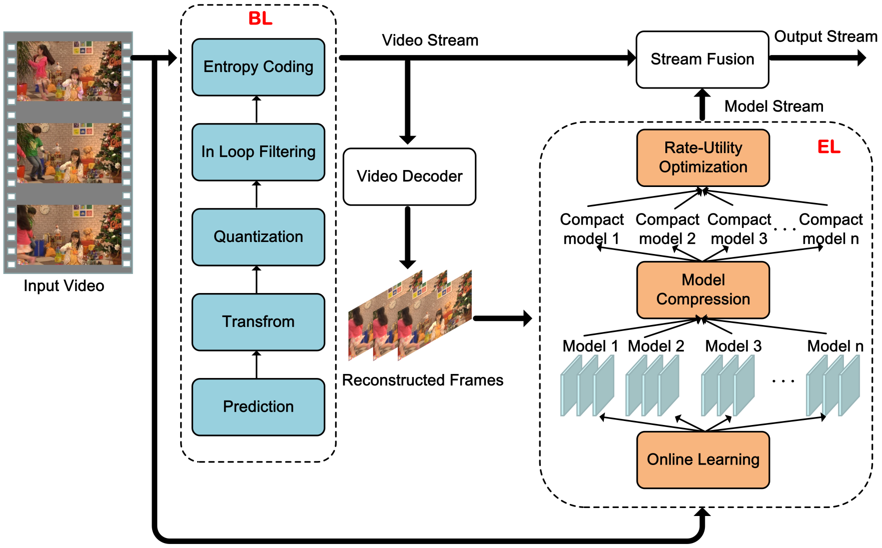


Figure 1: Illustration of the framework of Modality Transferable Visual Information Representation.

networks [32, 41–43, 54, 81] focus on the improvement of intra prediction efficiency by creating more powerful intra prediction modes. Deep neural network based inter prediction methods have improved the prediction efficiency [34, 73, 82, 83] by generating a more convincing prediction. Song *et al.* [60] proposed a deep neural network based entropy coding method to directly predict the probability distribution of intra modes instead of relying on the handcrafted context models, such that the statistical redundancy can be further removed, leading to higher coding efficiency. DNNs based loop filtering methods have been widely studied [22, 36, 37, 49, 53, 80], which learn the mapping between the original patches and the degraded patches to eliminate the inevitable distortion introduced by the block-based hybrid video compression framework.

In addition, there are a series of approaches incorporating an adaptively learned CNN model in in-loop filtering during the standardlization of VVC [31, 77]. Instead of online learning an adaptive model only, our method aims to explore the capability of modality transferable visual information representation with the design philosophy of rate-utility optimization. Moreover, with the proposed rate-utility optimization, the modality transfer capability can be better exploited by incorporating the proposed method with any compact representation frameworks. In view of this, we incorporate the proposed scheme with the state-of-the-art VVC codec as an enhancement layer, and superior coding performance has been achieved. It is noteworthy that the enhancement layer in our method is CNN based and applied on the degraded frames of base layer, such that the proposed scheme is compared with these CNN based video quality enhancement methods on VVC.

3 MODALITY TRANSFERABLE VISUAL INFORMATION REPRESENTATION

3.1 Framework

As illustrated in Fig. 1, the proposed Modality Transferable Visual Information Representation (MTVIR) framework is composed of Base Layer (BL) and Enhancement Layer (EL). Specifically, the BL aligns with the traditional video codec, including several modules (prediction, transform, quantization, in loop filtering, and entropy coding) which are used to produce compact representation of visual signals based on the removal of intrinsic redundancies. The signal divergences between the original and distorted videos inevitably introduced in the BL are compensated with the modality of deep neural network which is specifically learned by the adaptive transferring of signal level distortion.

As such, the EL is introduced, which is composed of three sequential modules including online learning, model compression and rate-utility optimization, and the compact representation of deep neural network is combined with the BL to form the final output stream. The framework is able to shift the signal representation to deep learning model representation which is essentially a compact model with enhanced generative capability. Moreover, increased degree of scalability and flexibility is also supported based on the scalable architecture, as the bitstream composed of BL and EL can be adaptively shaped according to the network and storage constrains, and at the decoder side the individual decoding of the BL already supports the reconstruction of the fine texture.

3.2 Online learning

Online learning in EL serves as the transferable engine to characterize and absorb the statistical divergences between the pristine signals and the distorted version, in an effort to transform the visual signals into a well established and highly compact model. To this end, a neural network is utilized to learn such a mapping. Towards a compact representation, the quantization is performed on the parameters of the neural network with scale transform and affine transform for achieving more compact representation.

In this work, the neural network model is redesigned based upon Squeeze-and-Excitation Filtering CNN (SEFCNN) [22], as shown in Fig. 2. In particular, we incorporate the mechanism of compact

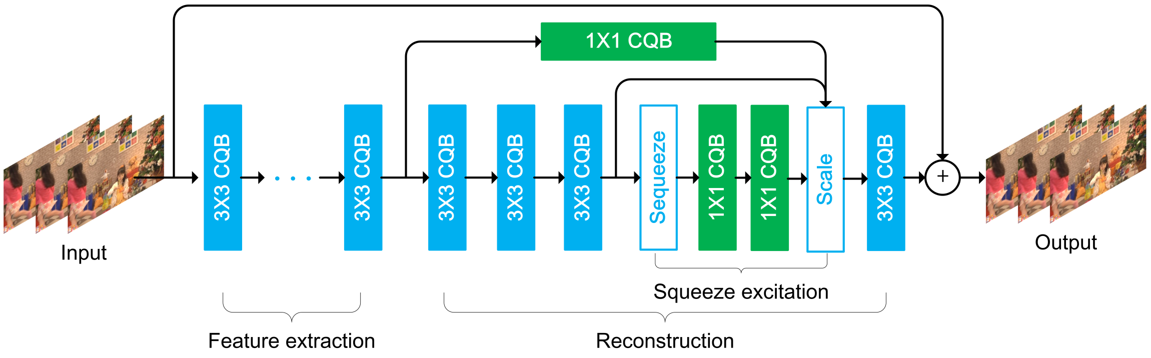


Figure 2: Diagram of the network structure.

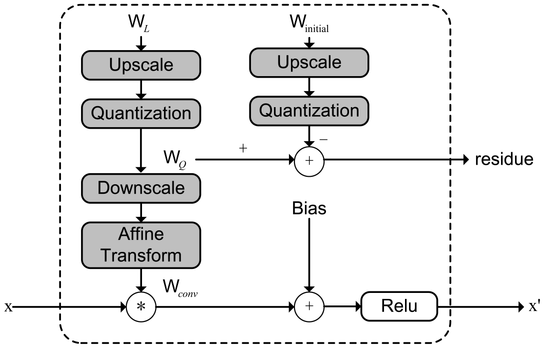


Figure 3: Illustration of the Conv Quantization Block.

Table 1: Parameter number comparisons between weights and biases.

Index	Weights	Biases	Total
Parameter number	40224	337	40561
Proportion	99.17%	0.83%	100%

Table 2: Parameter setting of biases.

Index	$bias^1, bias^2, ..., bias^{21}$	$bias^{22}$
Parameter number	1×16×21	1

neural network representation in this model by leveraging the scale transform, quantization operation and affine transform which are integrated as the Conv Quantization Block (CQB), as shown in Fig. 3. In particular, x and x' represent the input signal and output signal of CQB, respectively. As such, instead of the convolutional weights and biases learned in SEFCNN, the to-be-learned parameters include the reparameterizations of convolutional weights which are subjected to be quantized, the biases as well as the weights of affine transform. More specifically, the scale transform, quantization and affine transform can be formulated as follows,

$$W_Q = round\left(W_L \times S_c\right), \tag{1}$$

and

$$W_{conv} = f_{\varphi}\left(W_Q \times \frac{1}{S_c}\right), \tag{2}$$

where W_Q indicates the quantized weights, W_L indicates the weights that should be learned, S_c indicates the scale factor, W_{conv} represents the convolutional weights in network, and $f_{\varphi}(\cdot)$ represents the affine transform. As can be seen, the quantized weights are obtained by rounding the scaled to-be-learned weights. Then the quantized weights are down-scaled and fed to affine transform to generate the convolutional weights. The residue between quantized model and initial quantized model will be compressed by arithmetic coding, which will be discussed in subsection 3.3.

To reduce training computational complexity and enable the residual transmission, the network is trained by fine-tuning through an initial model in online learning. More specifically, an initial model with our architecture is obtained by using a large scale dataset and then the online models are learned specifically to fit the statistics of the video signals. Since the quantization operation in the network is indifferentiable, the "straight-through" gradient estimator proposed by Bengio *et al.* [8] is adopted, which performs forward rounding and backpropagates the gradient directly through the quantization operation to make the network trainable. In our method, Mean Squared Error (MSE) is adopted as the loss function to pursue the minimization on the difference between the output of the network and the ground truth. The to-be-learned weights are updated by minimizing the MSE loss, and subsequently the quantized weights and convolutional weights can be obtained. Our network is trained with limited frames, such that the time consumption is controllable to a certain extent.

3.3 Model compression

Model compression aims to compactly represent the learned model by exploiting both intra model redundancy and inter model redundancy. To remove the redundancy within a model, the quantization of the to-be-learned weights is performed, as detailed in subsection 3.2, in an effort to reduce the model size for representation. To remove the redundancy across models, the residue between current learned quantized model and the reference quantized model which is universally initialized is encoded to further shrink the model stream.

Table 3: Parameter setting of weights.

Index	W_{conv}^{1}	W_{conv}^{2}, ..., W_{conv}^{18}	W_{conv}^{19}, W_{conv}^{20}, W_{conv}^{21}	W_{conv}^{22}
Receptive field	3×3	3×3	1×1	3×3
Feature map number	16	16	16	16
Parameter number	3×3×1×16	3×3×16×16×17	1×1×16×16×3	3×3×16×1

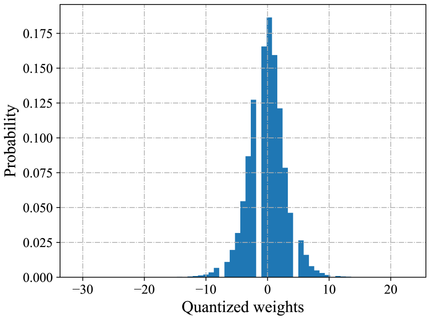


Figure 4: The distribution of quantized weights at 1000 iterations.

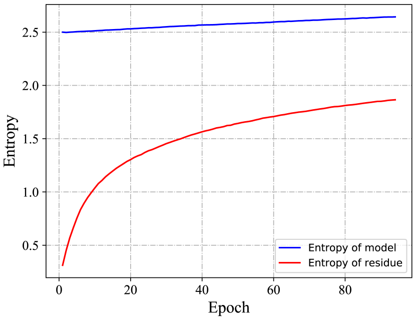


Figure 5: Entropy comparisons between the original model and residue.

Regarding the redundancy removal within a model, we apply quantization to represent the neural network model in a quantized form. This greatly facilitates efficient compression on these discrete values with ensured performance, since the quantization operation has already been embedded in the network during the training process. In our method, the parameter number of W_Q is the same as W_L and W_{conv}. We only quantize the reparameterizations of convolutional weights which account for a large proportion of the model representation, as illustrated in Tables 1, 2 and 3. The number of parameters influences the model size and performance. The model with more filters achieves better recovery performance with the expense of higher representation overhead. As such, a trade-off between number of parameters and the recovery performance is expected. We adopt the 32-bit floating point format to represent W_L for accurate learning while the quantized parameters W_Q are in a 32-bit integer format.

In theory, the number of bits for representing W_L and W_Q is identical, while practically the average number of bits can be reduced by the quantization process. In a common sense, the average number of bits for representing 32-bit floating number is 32. The underlying reason is that the distribution of floating number W_L during training is continuous, and it is quite rare that two floating numbers share the same value. However, the distribution of the quantized weights W_Q is discrete as shown in Fig. 4 and the average number of bits for representation depends on the entropy of the discrete signal. To quantitatively quantify the reduction of redundancy within a model by quantization, the entropy of the quantized parameters W_Q at 30000 iterations is calculated, which is 2.31 while the number of bits to represent W_L is 32. In this manner, it is estimated that the compression ratio of 13.85 times can been achieved, taking the sequence of “BQTerrace” as an example.

Regarding the redundancy existing between different models, the residue between online learned quantized model and initial quantized model is compressed by arithmetic coding, which can further economize the model transmission cost and achieve better compression efficiency. The inherent reason lies in the fact that weights of current model are learned by fine-tuning the initial model such that there exist high correlations. To illustrate the efficiency of removing the inter model redundancies, the entropy of residue and quantized parameters W_Q are presented in Fig. 5. It can be found that the entropy of residue is increasing with the epoch due to larger difference between learned model at each iteration and the initial model. It is also interesting to see that the entropy of residue is increasing fast at the beginning of online learning and converges after a few epochs (here we set 1000 iterations as one epoch). Moreover, the entropy of the learned model is much higher than the corresponding residue at each iteration. The residue compressed by arithmetic coding is transmitted along with the BL to the receiver side.

To illustrate the model compression performance, the compression ratio is further calculated at each epoch. It is worth mentioning that the biases and the weights of affine transform are not quantized, and the raw values in float32 format are incorporated into the final bitstream. Therefore, the total model stream includes the compressed residue of quantized weights, the biases and the weights of affine transform. More specifically, the fixed size of biases and weights of affine transform is 1.348KB and 0.368KB, respectively. The compression ratio is formulated as follows,

$$Ratio = \frac{R_{ori}}{R_{model}}, \tag{3}$$

where

$$R_{model} = R_{res} + R_{Biases} + R_{f_{\varphi}}. \tag{4}$$

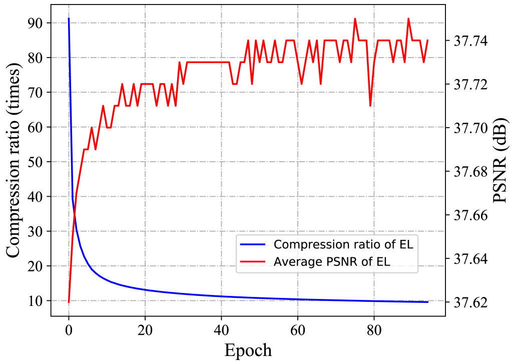


Figure 6: Compression ratio *vs* PSNR.

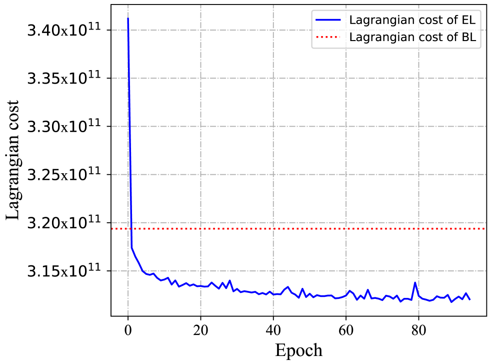


Figure 7: Lagrangian cost during online learning.

Here, R_{ori} and R_{model} indicate the number of bits of the original uncompressed model and the compressed model, respectively. R_{res}, R_{Biases} and R_{f_φ} indicate the number of bits of residue, biases and the weights of affine transform, respectively.

As shown in Fig. 6, at the beginning of online learning, we can achieve relatively high compression ratio, due to the fact that the difference between the current model and initial model is relatively small. With the decrease of compression ratio, the recovery capability becomes better in the early stage of online learning. However, after several epochs, both the compression ratio and performance of recovery converge. Although there are models with promising performance by matching the original signals, the final performance governed by both recovery capability and overhead of EL may not be satisfactory. Consequently, the selection of models becomes critical, which further motivates us to design the model selection scheme based on rate-utility optimization.

3.4 Rate-utility optimization

Rate-utility optimization aims to figure out an optimally compressed model that achieves the best trade-off between model rate and utility. Herein, the utility is defined based on the final utility of the EL, i.e., recovering the visual signal. As such, instead of the distortion of the compressed model, the quality of visual signals is what matters. In our scheme, after model compression, the candidate compressed models are obtained, which are subjected to be evaluated by the Lagrangian cost,

$$J = \sum_{i=1}^{N} J_i, \tag{5}$$

and

$$J_i = D_i + \lambda_i \times (R_i + \frac{R_{model}}{N}), \tag{6}$$

where J_i indicates the Lagrangian cost of i^{th} frame after enhancement and N is the number of frames in the group. D_i indicates the Sum of the Squared Error (SSE) of i^{th} enhanced frame by the selected model, λ_i indicates the Lagrange multiplier of i^{th} frame. R_i indicates the number of bits of i^{th} frame encoded in BL, and R_{model} is the number of bits of selected model. It is worth mentioning that the cost of model rate is assigned to each frame. Finally, the model with the minimum Lagrangian cost is selected and forms as the EL.

Table 4: Performance comparisons in terms of BD-Rate under AI configuration (anchor: VTM-4.0).

Sequences	JVET-N0254	JVET-N0513	Proposed
Class A	-1.22%	-0.39%	-1.94%
Class B	-0.93%	-0.58%	-2.51%
Class C	-1.90%	-1.63%	-3.65%
Class D	-2.57%	-1.32%	-2.08%
Class E	-2.22%	-2.15%	-5.34%
Class F	-1.09%	-1.00%	-4.90%
Overall	-1.56%	-1.09%	-2.63%

Taking the sequence of "BQTerrace" as an example, the Lagrangian cost of model at each epoch is shown in Fig. 7. The red line represents the Lagrangian cost of the BL and the first point of the curve indicates the Lagrangian cost of the initial model. It can be found that in this case, the Lagrangian cost decreases with the increasing of the epoch, finally leading to better representation performance. It is also interesting to see that the cost of the initial model is worse than the BL due to the fact that the recovery performance may not always be satisfactory when the adaptation to the specific content is absence.

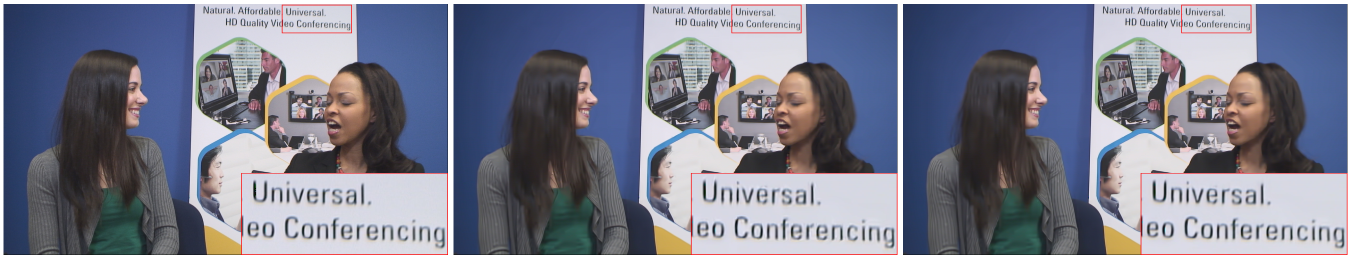
4 EXPERIMENTAL RESULTS AND ANALYSES

4.1 Experimental setup

To evaluate the performance of our method, the reference software of the upcoming video coding standard VVC (VVC Test Model version 4.0, VTM-4.0) is incorporated as the BL. The initial model is trained by using the database of DIV2K [2], which consists of 900 PNG pictures with the resolution of 2K (800 images for training and 100 images for validation). To facilitate the comparison with other methods, the video sequences are compressed with All Intra (AI) and Random Access (RA) configurations under Common Test Conditions (CTC) [10]. The learned models are applied on the luminance channel and the Quantization Parameters (QPs) are set following CTC {22, 27, 32, 37}.

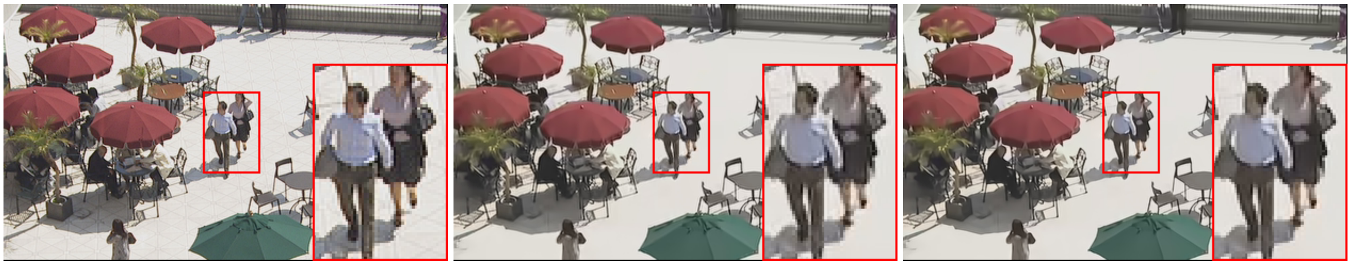
Table 5: Performance comparisons in terms of BD-Rate under RA configuration (anchor: VTM-4.0).

Sequences	JVET-N0110	JVET-N0254	JVET-N0480	JVET-N0513	Proposed
Class A	-2.21%	-1.74%	-1.06%	-0.37%	-3.21%
Class B	-1.52%	-1.13%	-0.55%	-0.43%	-4.64%
Class C	0.12%	-1.39%	0.09%	-0.76%	-4.60%
Class D	-	-1.39%	-	-0.79%	-4.50%
Class F	-	-0.50%	-	-0.35%	-3.70%
Overall	-1.36%	-1.27%	-0.58%	-0.52%	-4.07%


Natural. Affordable. Universal.
HD Quality Video Conferencing
Universal.
eo Conferencing

(a) Original (b) VVC (37.65 dB) (c) Proposed (38.18 dB)

Figure 8: Visual quality comparisons for "KristenAndSara" under AI configuration, where the 241-th frame is shown (QP=37).



(a) Original (b) VVC (30.40 dB) (c) Proposed (31.32 dB)

Figure 9: Visual quality comparisons for "BQSquare" under RA configuration, where the 184-th frame is shown (QP=37).

The module of online learning in EL is implemented with tensorflow software package [1]. All the frames of input video and corresponding reconstructed frames are cropped into patches randomly and the patch size is set as 35 × 35. The learning rate is set as 0.0002. It should be noted that the online learning will terminate if the size is larger than 13.5KB. The scale factor in scale transform is set as 10.

4.2 Performance comparisons

In this section, the proposed method is compared with the state-of-the-art algorithms that enhance the quality of decoded videos on the platform of VVC, including JVET-N0110 [31], JVET-N0254 [67], JVET-N0480 [77] and JVET-N0513 [21]. The coding performance is measured by BD-Rate [9] and the anchor is VTM 4.0. In particular, in JVET-N0254 and JVET-N0513, a dense residual CNN and two light weight deep CNNs are learned by the offline learning scheme. The performance of the offline learning based methods highly depends on the training dataset while our proposed method can well adapt to the variation on video content. In JVET-N0110 and JVET-N0480, online learning scheme is utilized to achieve an adaptive CNN loop filter and the parameters of the learned model are signaled to the decoder side. Compared to their methods, in our scheme the model compression is naturally incorporated in the learning process, and the final representation of the model is selected in a rate-utility optimization sense, leading to the improvement of the performance.

From Table 4, it is observed that the proposed method achieves 2.63% bit-rate savings on average under AI configuration while the methods of JVET-N0254 and JVET-N0513 achieve 1.56% and 1.09% bit rate reductions, respectively. From Table 5, our proposed method achieves 4.07% bit-rate savings on average under RA configuration while the methods of JVET-N0110, JVET-N00254, JVET-N0480, and JVET-N0513 reduce 1.36%, 1.27%, 0.58% and 0.52% bit rate on average, respectively. It can be easily found that our proposed method

Table 6: The coding performance and corresponding model bitrate for each sequence of CLASS B under RA configuration (anchor: VTM-4.0).

Sequences	Frame rate	QP	Bitrate (VTM-4.0) (Kb/s)	Model size (KB)	Model bitrate (Kb/s)	Δ PSNR (dB)	BD-Rate
MarketPlace	60	22	14252.95	10.95	8.76	0.04	-1.73%
		27	5426.06	12.06	9.65	0.05	
		32	2430.38	10.77	8.61	0.06	
		37	1079.80	12.22	9.77	0.07	
RitualDance	60	22	9443.66	11.16	8.93	0.10	-2.20%
		27	4717.00	11.39	9.11	0.12	
		32	2514.74	11.01	8.80	0.12	
		37	1322.51	11.20	8.96	0.12	
Cactus	50	22	14402.89	12.49	9.99	0.05	-4.48%
		27	4300.21	12.13	9.71	0.09	
		32	1998.80	12.56	10.06	0.12	
		37	971.69	12.58	10.06	0.16	
BasketballDrive	50	22	14684.65	12.05	9.64	0.04	-4.33%
		27	4788.42	12.25	9.80	0.09	
		32	2235.44	12.23	9.79	0.13	
		37	1117.68	12.48	9.98	0.15	
BQTerrace	60	22	34653.52	13.06	10.45	0.10	-10.45%
		27	5827.09	12.73	10.18	0.15	
		32	1765.12	12.37	9.89	0.18	
		37	779.22	12.27	9.82	0.22	

outperforms these CNN based algorithms under both AI and RA configurations. In Table 4, it is also observed that the performance of class D of our proposed is marginally worse than the method of JVET-N0254. The reason is that the resolution of sequences in class D is 416 × 240, which is smaller than that of other sequences. As such, the relatively smaller bit rates of video streams in BL lead to higher percentage of the overhead bit rate in EL. Moreover, under AI configuration only one frame in a group (8 frames) is encoded such that the relative overhead bit rate (model stream) is much higher than RA configuration even the size of model is very close. However, as shown in Table 5, our method outperforms other methods in all classes under RA configuration because the overhead bit rate in EL is assigned to all frames in a group. To further demonstrate the relationship between PSNR gain and the bit rate in EL, the results of each sequence in class B are presented in Table 6. It can be found that for the same sequence, the model size is different under different QP settings since our method selects the optimal model with minimum rate-utility cost.

Regarding the subjective quality comparisons, the original frames, VVC decoded frames, and reconstructed frames from the proposed method for "KristenAndSara" and "BQSquare" sequences are shown in Figs. 8 and 9. For better visualization, certain regions are also enlarged. It can be observed that the ringing artifacts and blocking artifacts are eliminated when compared with the anchor. The degraded structural details have also been well recovered since our method can well leverage the deep neural network representation to accommodate the statistics of the visual signals. More specifically, when enlarging Figs. 8 (b) and (c), it can be observed the marginal pixels of "U" in (b) are mixed with unexpected white pixels of background. In (c), these white pixels are suppressed. In Figs. 9 (b) and (c), scrupulous observers may also find that the area around the woman's left hand in (c) is smoother than the area in (b). Due to powerful transferable capability with acceptable rate overhead, the proposed method achieves effectively improved visual information representation performance.

5 CONCLUSIONS

In this paper, a novel scheme for visual signal representation that leverages transferable modality has been proposed. In particular, online learning that accommodates the statistics of input signals serves as a transferable engine from visual information to neural network model which is further compactly represented via inter/intra model reduduancy removal. The trade-off between rate and utility is further optimized, leading to the best representation capability. With the state-of-the-art video coding platform of VVC, extensive experiments show that visual information capability has been improved with significant bit rate savings.

ACKNOWLEDGMENTS

This work was supported in part by the Hong Kong RGC General Research Funds under Grant 9042322 (CityU 11200116), Grant 9042489 (CityU 11206317), and Grant 9042816 (CityU 11209819). This work was also supported in part by the Natural Science Foundation of China under Grant 61672443, Grant 61901459, and in part by China Postdoctoral Science Foundation under Grant 2019M653127.

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Eirikur Agustsson and Radu Timofte. 2017. NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*.

[3] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. 2019. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE International Conference on Computer Vision*. 221–231.

[4] Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep?. In *Advances in neural information processing systems*. 2654–2662.

[5] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. 2015. Density modeling of images using a generalized normalization transformation. *arXiv preprint arXiv:1511.06281* (2015).

[6] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. 2016. End-to-end optimization of nonlinear transform codes for perceptual quality. In *2016 Picture Coding Symposium (PCS)*. IEEE, 1–5.

[7] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. 2016. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704* (2016).

[8] Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432* (2013).

[9] Gisle Bjøntegaard. 2001. Calculation of average PSNR differences between RD-curves (VCEG-M33). In *VCEG Meeting (ITU-T SG16 Q. 6)*. 2–4.

[10] Frank Bossen, Jill Boyce, Karsten Suehring, Xiang Li, and Vadim Seregin. 2019. JVET common test conditions and software reference configurations for SDR video. *Joint Video Exploration Team (JVET), doc. JVET-N1010* (Mar. 2019).

[11] Benjamin Bross, Jianle Chen, and Shan Liu. 2018. Versatile Video Coding (Draft 3). *Joint Video Exploration Team (JVET), doc. JVET-L1001* (2018).

[12] Benjamin Bross, Jianle Chen, and Shan Liu. 2019. Versatile Video Coding (Draft 4). *Joint Video Exploration Team (JVET), doc. JVET-M1001* (2019).

[13] Benjamin Bross, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. 2018. CE3: Multiple reference line intra prediction. *Joint Video Exploration Team (JVET), doc. JVET-K0051* (Jul. 2018).

[14] Yue Chen, Debargha Murherjee, Jingning Han, Adrian Grange, Yaowu Xu, Zoe Liu, Sarah Parker, Cheng Chen, Hui Su, Urvang Joshi, et al. 2018. An overview of core coding tools in the AV1 video codec. In *2018 Picture Coding Symposium (PCS)*. IEEE, 41–45.

[15] Ziqian Chen, Ling-Yu Duan, Shiqi Wang, Yihang Lou, Tiejun Huang, Dapeng Oliver Wu, and Wen Gao. 2019. Toward Knowledge as a Service Over Networks: A Deep Learning Model Communication Paradigm. *IEEE Journal on Selected Areas in Communications* 37, 6 (2019), 1349–1363.

[16] Ziqian Chen, Shiqi Wang, Dapeng Oliver Wu, Tiejun Huang, and Ling-Yu Duan. 2018. From data to knowledge: Deep learning model compression, transmission and communication. In *Proceedings of the 26th ACM international conference on Multimedia*. 1625–1633.

[17] Yu Cheng, Duo Wang, Pan Zhou, and Tao Zhang. 2017. A survey of model compression and acceleration for deep neural networks. *arXiv preprint arXiv:1710.09282* (2017).

[18] Narae Choi, Yinji Piao, Kiho Choi, and Chanyul Kim. 2018. CE3.3 related: Intra 67 modes coding with 3 MPM. *Joint Video Exploration Team (JVET), doc. JVET-K0529* (Jul. 2018).

[19] Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. Binaryconnect: Training deep neural networks with binary weights during propagations. In *Advances in neural information processing systems*. 3123–3131.

[20] Matthieu Courbariaux, Itay Hubara, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830* (2016).

[21] Yuanying Dai, Dong Liu, Ning Yan, and Feng Wu. 2019. CE13: Experimental results of CNN-based In-Loop Filter (USTC). *Joint Video Exploration Team (JVET), doc. JVET-N0513* (Mar. 2019).

[22] Dandan Ding, Lingyi Kong, Guangyao Chen, Zoe Liu, and Yong Fang. 2019. A Switchable Deep Learning Approach for In-loop Filtering in Video Coding. *IEEE Transactions on Circuits and Systems for Video Technology* (2019).

[23] Xin Dong, Shangyu Chen, and Sinno Pan. 2017. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In *Advances in Neural Information Processing Systems*. 4857–4867.

[24] Wen Gao and Siwei Ma. 2014. An overview of AVS2 standard. *Advanced Video Coding Systems* 22 (Jan. 2014), 35–49.

[25] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir Bourdev. 2014. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115* (2014).

[26] Yiwen Guo, Anbang Yao, and Yurong Chen. 2016. Dynamic network surgery for efficient dnns. In *Advances In Neural Information Processing Systems*. 1379–1387.

[27] Song Han, Huizi Mao, and William J Dally. 2015. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149* (2015).

[28] Yihui He, Xiangyu Zhang, and Jian Sun. 2017. Channel pruning for accelerating very deep neural networks. In *Proceedings of the IEEE International Conference on Computer Vision*. 1389–1397.

[29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).

[30] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[31] Yu-Ling Hsiao, Olena Chubach, Ching-Yeh Chen, Chuang Tzu-Der, Chih-Wei Hsu, Yu-Wen Huang, and Shaw-Min Lei. 2019. CE13-1.1: Convolutional neural network loop filter. *Joint Video Exploration Team (JVET), doc. JVET-N0110* (Mar. 2019).

[32] Yueyu Hu, Wenhan Yang, Sifeng Xia, Wen-Huang Cheng, and Jiaying Liu. 2018. Enhanced intra prediction with recurrent neural network in video coding. In *2018 Data Compression Conference*. IEEE, 413–413.

[33] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks. In *Advances in neural information processing systems*. 4107–4115.

[34] Shuai Huo, Dong Liu, Feng Wu, and Houqiang Li. 2018. Convolutional neural network-based motion compensation refinement for video coding. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 1–4.

[35] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360* (2016).

[36] Chuanmin Jia, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, Jiaying Liu, Shiliang Pu, and Siwei Ma. 2019. Content-aware convolutional neural network for in-loop filtering in high efficiency video coding. *IEEE Transactions on Image Processing* 28, 7 (2019), 3343–3356.

[37] Chuanmin Jia, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, and Siwei Ma. 2017. Spatial-temporal residue network based in-loop filter for video coding. In *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 1–4.

[38] Yong-Deok Kim, Eunhyeok Park, Sungjoo Yoo, Taelim Choi, Lu Yang, and Dongjun Shin. 2015. Compression of deep convolutional neural networks for fast and low power mobile applications. *arXiv preprint arXiv:1511.06530* (2015).

[39] Thorsten Laude, Yannick Richter, and Jörn Ostermann. 2018. Neural Network Compression using Transform Coding and Clustering. *arXiv preprint arXiv:1805.07258* (2018).

[40] Cong Leng, Zesheng Dou, Hao Li, Shenghuo Zhu, and Rong Jin. 2018. Extremely low bit neural network: Squeeze the last bit out with admm. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

[41] Jiahao Li, Bin Li, Jizheng Xu, Ruiqin Xiong, and Wen Gao. 2018. Fully connected network-based intra prediction for image coding. *IEEE Transactions on Image Processing* 27, 7 (2018), 3236–3247.

[42] Yue Li, Li Li, Zhu Li, Jianchao Yang, Ning Xu, Dong Liu, and Houqiang Li. 2018. A hybrid neural network for chroma intra prediction. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 1797–1801.

[43] Yue Li, Dong Liu, Houqiang Li, Li Li, Feng Wu, Hong Zhang, and Haitao Yang. 2018. Convolutional neural network-based block up-sampling for intra frame coding. *IEEE Transactions on Circuits and Systems for Video Technology* 28, 9 (2018), 2316–2330.

[44] Darryl Lin, Sachin Talathi, and Sreekanth Annapureddy. 2016. Fixed point quantization of deep convolutional networks. In *International Conference on Machine Learning*. 2849–2858.

[45] Shaohui Lin, Rongrong Ji, Xiaowei Guo, Xuelong Li, et al. 2016. Towards Convolutional Neural Networks Compression via Global Error Reconstruction.. In *IJCAI*. 1753–1759.

[46] Christos Louizos, Karen Ullrich, and Max Welling. 201. Bayesian compression for deep learning. In *Advances in Neural Information Processing Systems*. 3288–3298.

[47] Ping Luo, Zhenyao Zhu, Ziwei Liu, Xiaogang Wang, and Xiaoou Tang. 2016. Face model compression by distilling knowledge from neurons. In *Thirtieth AAAI conference on artificial intelligence*.

[48] Marc Masana, Joost van de Weijer, Luis Herranz, Andrew D Bagdanov, and Jose M Alvarez. 2017. Domain-adaptive deep network compression. In *Proceedings of the IEEE International Conference on Computer Vision*. 4289–4297.

[49] Xiandong Meng, Chen Chen, Shuyuan Zhu, and Bing Zeng. 2018. A New HEVC In-Loop Filter Based on Multi-channel Long-Short-Term Dependency Residual Networks. In *2018 Data Compression Conference*. IEEE, 187–196.

[50] Debargha Mukherjee, Jingning Han, Jim Bankoski, Ronald Bultje, Adrian Grange, John Koleszar, Paul Wilkins, and Yaowu Xu. 2013. A technical overview of vp9–the latest open-source video codec. In *SMPTE 2013 Annual Technical Conference & Exhibition*. SMPTE, 1–17.

[51] Deniz Oktay, Johannes Ballé, Saurabh Singh, and Abhinav Shrivastava. 2019. Scalable Model Compression by Entropy Penalized Reparameterization. arXiv:cs.LG/1906.06624

[52] Eunhyeok Park, Junwhan Ahn, and Sungjoo Yoo. 2017. Weighted-entropy-based quantization for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 5456–5464.

[53] Woon-Sung Park and Munchurl Kim. 2016. CNN-based in-loop filtering for coding efficiency improvement. In *2016 IEEE 12th Image, Video, and Multidimensional Signal Processing Workshop (IVMSP)*. IEEE, 1–5.

[54] Jonathan Pfaff, Philipp Helle, Dominique Maniry, Stephan Kaltenstadler, Björn Stallenberger, Philipp Merkle, Mischa Siekmann, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. 2018. Intra prediction modes based on neural networks. *Joint Video Exploration Team (JVET), doc. JVET-J0037* (Apr. 2018).

[55] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*. Springer, 525–542.

[56] Brandon Reagen, Udit Gupta, Robert Adolf, Michael M Mitzenmacher, Alexander M Rush, Gu-Yeon Wei, and David Brooks. 2017. Weightless: Lossy weight encoding for deep neural network compression. *arXiv preprint arXiv:1711.04686* (2017).

[57] Oren Rippel and Lubomir Bourdev. 2017. Real-time adaptive image compression. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2922–2930.

[58] Shibani Santurkar, David Budden, and Nir Shavit. 2018. Generative compression. In *2018 Picture Coding Symposium (PCS)*. IEEE, 258–262.

[59] Wenling Shang, Kihyuk Sohn, Diogo Almeida, and Honglak Lee. 2016. Understanding and improving convolutional neural networks via concatenated rectified linear units. In *international conference on machine learning*. 2217–2225.

[60] Rui Song, Dong Liu, Houqiang Li, and Feng Wu. 2017. Neural network-based arithmetic coding of intra prediction modes in HEVC. In *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 1–4.

[61] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. 2012. Overview of the High Efficiency Video Coding (HEVC) Standard. *IEEE Transactions on circuits and systems for video technology* 22, 12 (2012), 1649–1668.

[62] Gary J Sullivan and Thomas Wiegand. 1998. Rate-distortion optimization for video compression. *IEEE signal processing magazine* 15, 6 (1998), 74–90.

[63] Vivienne Sze and Madhukar Budagavi. 2012. High Throughput CABAC Entropy Coding in HEVC. *IEEE Transactions on Circuits and Systems for Video Technology* 22, 12 (Dec 2012), 1778–1791. https://doi.org/10.1109/TCSVT.2012.2221526

[64] George Toderici, Sean M O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. 2015. Variable rate image compression with recurrent neural networks. *arXiv preprint arXiv:1511.06085* (2015).

[65] Michael Tschannen, Eirikur Agustsson, and Mario Lucic. 2018. Deep generative models for distribution-preserving lossy compression. In *Advances in Neural Information Processing Systems*. 5929–5940.

[66] Gregory K Wallace. 1990. Overview of the JPEG (ISO/CCITT) still image compression standard. In *Image Processing Algorithms and Techniques*, Vol. 1244. International Society for Optics and Photonics, 220–233.

[67] Yingbin Wang, Zhenzhong Chen, Yiming Li, Liang Zhao, Shan Liu, and Xiang Li. 2019. CE13:Dense Residual Convolutional Neural Network based In-Loop Filter (Test 2.2 and 2.3). *Joint Video Exploration Team (JVET), doc. JVET-N0254* (Mar. 2019).

[68] Yunhe Wang, Chang Xu, Chao Xu, and Dacheng Tao. 2017. Beyond filters: Compact feature map for portable deep model. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 3703–3711.

[69] Simon Wiedemann, Arturo Marban, Klaus-Robert Müller, and Wojciech Samek. 2019. Entropy-constrained training of deep neural networks. In *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 1–8.

[70] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. 2003. Overview of the H. 264/AVC video coding standard. *IEEE Transactions on circuits and systems for video technology* 13, 7 (2003), 560–576.

[71] Dapeng Wu, Zhenjiang Li, Jianping Wang, Yuanqing Zheng, Mo Li, and Qiuyuan Huang. 2017. Vision and challenges for knowledge centric networking (KCN). *arXiv preprint arXiv:1707.00805* (2017).

[72] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. 2016. Quantized convolutional neural networks for mobile devices. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4820–4828.

[73] Ning Yan, Dong Liu, Houqiang Li, Bin Li, Li Li, and Feng Wu. 2018. Convolutional neural network-based fractional-pixel motion compensation. *IEEE Transactions on Circuits and Systems for Video Technology* 29, 3 (2018), 840–853.

[74] Jianbo Ye, Xin Lu, Zhe Lin, and James Z Wang. 2018. Rethinking the smaller-norm-less-informative assumption in channel pruning of convolution layers. *arXiv preprint arXiv:1802.00124* (2018).

[75] Hyunho Yeo, Sunghyun Do, and Dongsu Han. 2017. How will deep learning change internet video delivery?. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*. 57–64.

[76] Hyunho Yeo, Youngmok Jung, Jaehong Kim, Jinwoo Shin, and Dongsu Han. 2018. Neural adaptive content-aware internet video delivery. In *13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18)*. 645–661.

[77] Hujun Yin, Rongzhen Yang, Xiaoran Fang, and Shoujiang Ma. 2019. CE13-1.2: Adaptive convolutional neural network loop filter. *Joint Video Exploration Team (JVET), doc. JVET-N0480* (Mar. 2019).

[78] Xiyu Yu, Tongliang Liu, Xinchao Wang, and Dacheng Tao. 2017. On compressing deep models by low rank and sparse decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7370–7379.

[79] Shuangfei Zhai, Yu Cheng, Zhongfei Mark Zhang, and Weining Lu. 2016. Doubly convolutional neural networks. In *Advances in neural information processing systems*. 1082–1090.

[80] Yongbing Zhang, Tao Shen, Xiangyang Ji, Yun Zhang, Ruiqin Xiong, and Qionghai Dai. 2018. Residual highway convolutional neural networks for in-loop filtering in HEVC. *IEEE Transactions on Image Processing* 27, 8 (2018), 3827–3841.

[81] Zheng-Teng Zhang, Chia-Hung Yeh, Li-Wei Kang, and Min-Hui Lin. 2017. Efficient CTU-based intra frame coding for HEVC based on deep learning. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 661–664.

[82] Lei Zhao, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. 2018. Enhanced ctu-level inter prediction with deep frame rate up-conversion for high efficiency video coding. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 206–210.

[83] Lei Zhao, Shiqi Wang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. 2019. Enhanced motion-compensated video coding with deep virtual reference frame generation. *IEEE Transactions on Image Processing* 28, 10 (2019), 4832–4844.

[84] Aojun Zhou, Anbang Yao, Yiwen Guo, Lin Xu, and Yurong Chen. 2017. Incremental network quantization: Towards lossless cnns with low-precision weights. *arXiv preprint arXiv:1702.03044* (2017).