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 compression; rate-utility optimization

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...
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
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 standardization 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
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,

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

and

$$W_{\text{conv}} = f_{\phi} \left( W_Q \times \frac{1}{S_c} \right),$$  

where $W_Q$ indicates the quantized weights, $W_L$ indicates the weights that should be learned, $S_c$ indicates the scale factor, $W_{\text{conv}}$ represents the convolutional weights in network, and $f_{\phi}(\cdot)$ 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 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 | $1 \times 16 \times 21$ | 1 |


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 \( W_{\text{conv}} \). 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,

\[
\text{Ratio} = \frac{R_{\text{ori}}}{R_{\text{model}}}, \quad (3)
\]

where

\[
R_{\text{model}} = R_{\text{res}} + R_{\text{Biases}} + R_{\text{fp}}. \quad (4)
\]

![Figure 4: The distribution of quantized weights at 1000 iterations.](image)

![Figure 5: Entropy comparisons between the original model and residue.](image)
Here, $R_{ori}$ and $R_{model}$ indicate the number of bits of the original uncompressed model and the compressed model, respectively. $R_{res}$, $R_{bias}$, and $R_{a}$ 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,

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

and

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

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, $\lambda_i$ 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.

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].

| 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%   |
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%   |

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

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 $35 \times 35$. 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       |         |
|              |            | 22 | 9443.66                  | 11.16           | 8.93                 | 0.10       |         |
| RitualDance  | 60         | 27 | 4717.00                  | 11.39           | 9.11                 | 0.12       | -2.20%  |
|              |            | 32 | 2514.74                  | 11.01           | 8.80                 | 0.12       |         |
|              |            | 37 | 1322.51                  | 11.20           | 8.96                 | 0.12       |         |
|              |            | 22 | 14402.89                 | 12.49           | 9.99                 | 0.05       |         |
| Cactus       | 50         | 27 | 4300.21                  | 12.13           | 9.71                 | 0.09       | -4.48%  |
|              |            | 32 | 1998.80                  | 12.56           | 10.06                | 0.12       |         |
|              |            | 37 | 971.69                   | 12.58           | 10.06                | 0.16       |         |
|              |            | 22 | 14684.65                 | 12.05           | 9.64                 | 0.04       |         |
| BasketballDrive | 50      | 27 | 4788.42                  | 12.25           | 9.80                 | 0.09       | -4.33%  |
|              |            | 32 | 2235.44                  | 12.23           | 9.79                 | 0.13       |         |
|              |            | 37 | 1117.68                  | 12.48           | 9.98                 | 0.15       |         |
|              |            | 22 | 34653.52                 | 13.06           | 10.45                | 0.10       |         |
| BQTerrace    | 60         | 27 | 5827.09                  | 12.73           | 10.18                | 0.15       | -10.45% |
|              |            | 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 $416 \times 240$, 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 $\text{âÂ¡\text{U}âÂ¡}$ 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 redundancy 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.
