IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 13, 2020, accepted August 25, 2020, date of publication August 31, 2020, date of current version September 17, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3020388

Neural Network-Based Video Compression Artifact Reduction Using Temporal Correlation and Sparsity Prior Predictions

WEI-GANG CHEN[1], RUNYI YU[2], (Senior Member, IEEE), AND XUN WANG[1], (Member, IEEE)

[1]School of Computer and Information Engineering, Zhejiang Gongshang University, Hangzhou 310018, China
[2]Department of Electrical and Electronic Engineering, Eastern Mediterranean University, 99628 Mersin, Turkey

Corresponding author: Wei-Gang Chen (wgchen_gsu@mail.zjgsu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61672460, in part by the Public Welfare Technology Research Project of Zhejiang Province under Grant LGG20F020005, and in part by the Science and Technology Program of Zhejiang Province (Key Research and Development Plan) under Grant 2020C01049.

ABSTRACT Quantization in lossy video compression may incur severe quality degradation, especially at low bit-rates. Developing post-processing methods that improve visual quality of decoded images is of great importance, as they can be directly incorporated in any existing compression standard or paradigm. We propose in this article a two-stage method, a texture detail restoration stage followed by a deep convolutional neural network (CNN) fusion stage, for video compression artifact reduction. The first stage performs in a patch-by-patch manner. For each patch in the current decoded frame, one prediction is formed based on the sparsity prior assuming that natural image patches can be represented by sparse activation of dictionary atoms. Under the temporal correlation hypothesis, we search the best matching patch in each reference frame, and select several matches with more texture details to tile motion compensated predictions. The second stage stacks the predictions obtained in the preceding stage along with the decoded frame itself to form a tensor, and proposes a deep CNN to learn the mapping between the tensor as input and the original uncompressed image as output. Experimental results demonstrate that the proposed two-stage method can remarkably improve, both subjectively and objectively, the quality of the compressed video sequence.

INDEX TERMS Compression artifact reduction, convolutional neural networks, high efficiency video coding, sparse representation, temporal correlation.

I. INTRODUCTION

Quantization in lossy image and video compression is a many-to-one mapping. This means that the decoded block can be quite different from the original one, especially at low bit-rates. Developing post-processing methods that improve visual quality of decoded images at decoder sides has attracted great interest of researchers, as they can be directly incorporated in any existing compression standard or paradigm. Most existing such methods can be classified into three categories. Methods in the first class are deblocking oriented, in which blocking artifacts are regarded as artificial discontinuities around block boundaries, and these annoying boundary pixels are smoothed out with linear or non-linear filters in spatial or frequency domain [1], [2]. Methods in the second class are restoration oriented. They regard the image compression as distortion, and the restoration from a decoded image is usually formulated as an ill-posed image inverse problem which is typically solved by exploiting some image model priors, including projections onto convex sets [3], [4], block-based sparse representation [5]–[11], total variation [12], and Markov random field [13], [14]. Recently, inspired by the great success of deep learning in other computer vision tasks such as single image super-resolution [15]–[17] and image denoising [18], researchers began to use deep convolutional neural networks (CNNs) for compression artifact reduction [19]–[25], leading to the third class, the CNNs based methods. In using deep neural networks one needs to train and learn a mapping function that estimates for a given decoded input its corresponding original

The associate editor coordinating the review of this manuscript and approving it for publication was Junxiu Liu.

uncompressed counterpart. Works originally developed for JPEG images typically use a single image as the input [19]–[21]. Direct extension to the video domain is possible in a frame-by-frame manner. However, the quality of a compressed video can fluctuate dramatically across frames [24]. In particular, for low quality frames, though the trained network may improve their visual quality to some degree, there would always exhibit a large number of visually noticeable artifacts. Therefore, works developed for image sequences typically use multiple frames as the input [23], [24], [26] to the CNN. With the help of its neighboring high-quality frames, a low-quality frame can be adequately enhanced.

Considering that compression artifact reduction is a severely under-determined inverse problem, we make two hypotheses in this study. The temporal correlation hypothesis assumes that for every patch in the current decoded frame, there exist several similar patches with more image details along the motion trajectory in neighboring frames; the sparsity prior assumes that natural image patches can be represented by sparse activations of dictionary atoms. Video compression artifact reduction is implemented by fusing the predictions obtained under these hypotheses, and proceeded on two stages: 1) texture detail restoration, and 2) deep CNN fusion. The first stage performs texture detail restoration in a patch-by-patch manner. For each patch in the current decoded frame, one prediction is estimated based on the sparsity prior, and another one or more are predicted based on the temporal correlation hypothesis. The second stage proposes a deep CNN to fuse the resulted predictions from the preceding stage.

The main contribution of this work is that it proposes a new method to address the challenge of restoring high frequency contents lost in quantization. The two hypotheses are utilized to construct multiple predictions. These predicted frames typically contain many regions that have more high frequency contents than the current decoded frame. As input of the deep CNN, these frames not only facilitate the network training, but also allow output of improved visual quality.

The remainder of this article is organized as follows. Section II reviews some previous studies related to compression artifact reduction. Section III describes our method for enhancing the visual quality of decoded videos. Experimental results are given in Section IV. Finally, Section V concludes our study.

II. RELATED WORK

As the concomitant of the integer transform and quantization in HEVC, the quantization error of a transform coefficient affects the pixels in the same block. In spatial domain, the quantization error for a given point is the sum of the errors of each transform coefficient multiplied by the corresponding 2-D DCT basis image [27]. The quantization process can also be considered as a low-pass filter. This means that a large amount of high-frequency components of the block are removed in the process. In view of this consideration, we regard the reduction of compression artifact as image detail restoration, and review the related work in two parts: 1) restoration oriented, and 2) deep learning methods for compression artifact reduction.

A. RESTORATION ORIENTED COMPRESSION ARTIFACT REDUCTION

Many restoration oriented methods have been proposed for compression artifact reduction. Some works rely on information in the DCT domain. Foi *et al.* [28] proposed to perform hard-thresholding and empirical Wiener filtering in the shape-adaptive DCT domain, and to utilize clipped or attenuated DCT coefficients to reconstruct a local estimation of the image signal within an adaptive shape support. Zhang *et al.* proposed [29] to estimate the DCT coefficients of each block by adaptively fusing two predictions: the coefficients decoded from the compressed bitstream and the weighted average of the coefficients in nonlocal blocks. Dar *et al.* [30] considered a linear approximation of the nonlinear compression-decompression process and formulated the compression artifact reduction as a regularized inverse problem which was solved by the alternating direction method of multipliers [31]. Zhang *et al.* [32] utilized both the spatial and temporal correlation to form three predictions. The first prediction was constructed by inversely quantizing transform coefficients directly; the second was derived by representing each transform block with a temporal auto-regressive model along its motion trajectory; and the third was inferred with the original coefficients from similar blocks in non-local regions. Li *et al.* [33] focused on super-resolution (SR) of compressed images. After analyzing the correlation between the deblocking and SR sub-process, they proposed an iterative cascading framework, where a feedback route was constructed to send extra information extracted from the SR outcome back to the deblocking module, and thus helped produce further performance gains.

Recently, learning-based methods, particularly those sparse representation-based, are gradually becoming a better choice. For solving image inverse problems such as compression artifact reduction, single image super-resolution and image denoising, sparse representation models an image x as a linear system

$$x = \Phi\alpha \tag{1}$$

where matrix $\Phi \in \mathbb{R}^{n\times K}$, and vector $\alpha \in \mathbb{R}^K$. Each column in Φ serves as an atomic image, and thus Φ itself is a dictionary of atoms. Vector α is known to be sparse with only $k \ll K$ non-zero elements. Note that α is a representation of x under Φ and describes which atoms and what ''portions'' thereof are used for its construction [34]. Learning dictionaries, which are necessary for mapping a degraded patch to a visually more appealing one, from example images is essential for sparse representation. Jung *et al.* [5] proposed to iteratively train a general dictionary for image deblocking by the K-SVD algorithm, and to estimate an error threshold in the reconstruction stage for orthogonal matching pursuit according to the compression factor of the compressed image. Works in [6], [10] introduced the notion of group-based sparse

representation (GSR) by explicitly exploiting the intrinsic local sparsity and nonlocal self-similarity of natural images. The GSR searches similar patches in an image and organizes them as a group, which is further assumed to be represented by a few atoms of a self-adaptive dictionary. The reconstruction is then obtained as the solution of an ℓ_0 minimization which is solved via its ℓ_1 surrogates and the split Bregman based iterative algorithm. In [7], an image was first decomposed into low-frequency (LF) and high-frequency (HF) parts. The HF part was further decomposed into blocking and non-blocking components. A dictionary was learned using training samples extracted from the HF parts of the image, and was divided into two sub-dictionaries, corresponding to blocking components and non-blocking components. Liu *et al.* [9] learned two dictionaries of PCA bases, one in the DCT domain and the other in the pixel domain. In restoration, two locally adaptive sparse representations that jointly determined the restored patch were generated using these two dictionaries.

B. DEEP LEARNING FOR COMPRESSION ARTIFACT REDUCTION

More recently, CNNs have been applied to compression artifact reduction following their success in many other computer vision tasks. Dong *et al.* [19] proposed an artifact reduction CNN (AR-CNN), which was constructed by adding a feature enhancement layer in their previously developed super-resolution CNN (SRCNN), and reported to achieve more than 1dB PSNR (peak signal-to-noise ratio) improvement over original JPEG images. Cavigelli *et al.* [20] proposed a deep CNN for image compression artifact suppression (CAS-CNN). It was a 12-layer CNN with hierarchical skip connections and was trained with a multi-scale loss function. It was reported that CAS-CNN allows a boost of up to 1.79dB in PSNR over JPEG images. Guo *et al.* [21] proposed an one-to-many network for compression artifact reduction. Their model consists of two components: the proposal component and the measurement component. Taking a JPEG compressed image as input, the former outputs a number of artifact-free candidates; and the latter evaluate the output quality using three loss functions. It was reported that their one-to-many network could reconstruct multiple artifact-free candidates that were more favored by humans. Dai *et al.* [22] proposed to replace deblocking and SAO in HEVC intra coding with a variable-filter-size residual learning CNN (VRCNN). The VRCNN can reportedly provide 4.6% BD-rate reduction on average against the HEVC reference implementation. Soh *et al.* [23] proposed a deep artifact reduction temporal network (ARTN) consisting of three temporal branches. One branch takes the current decoded frame, and the other two take the motion compensated frames as input. The outputs of the three branches are then concatenated and fed to a single network. It was reported that this ARTN achieved about 0.23dB PSNR improvement over the conventional networks for HEVC artifact reduction. Guan *et al.* [24] observed that there typically exists large quality variation in consecutive frames of a compressed video. Therefore, it is possible to improve the quality of a low-quality frame with the help of its neighboring high-quality frames referred to as Peak Quality Frames (PQFs). To this end, they proposed a bidirectional long short-term memory (BiLSTM) model for no-reference PQF detection and a multi-frame CNN architecture for non-PQF quality enhancement. Galteri *et al.* [25] proposed an image transformation approach based on a feed-forward fully convolutional residual network. Their model can be optimized either directly in terms of an image similarity loss or using a generative adversarial network (GAN), and they showed via experiments that GAN is capable of producing higher quality images with sharp details.

III. THE PROPOSED TWO-STAGE METHOD

As shown in Figure 1, the proposed artifact reduction method proceeds on two stages, namely, 1) an texture detail restoration stage, and followed by 2) a deep CNN fusion stage. The first one performs image detail restoration in a patch-by-patch manner; whereas the second uses a deep CNN to fuse the predictions of the preceding stage.

In applying deep CNN to solve regression problems such as compression artifact reduction and single image super-resolution, higher-quality inputs not only improve the probability of the model in predicting better outputs, but also facilitate the network training. Recognizing the significant role of the input in affecting the performance of the deep CNN, we propose the two-stage method for compression artifact reduction. The first stage performs in a patch-by-patch manner, and aims to hallucinate the missing details of the patches. Based on the temporal correlation hypothesis, we search matching patches in reference frames for a given patch in the current frame. Patches with more texture details are selected to tile motion compensated predictions. Based on the sparsity prior, we learn dictionary pairs representing the mapping between HEVC-compressed patches and their corresponding ground-truths. In the second stage, a deep CNN, which takes a degraded frame and the multi-hypothesis based predictions of the first stage as the input, is constructed to estimate the artifact-free image.

A. PATCH PREDICTION USING TEMPORAL CORRELATION PRIOR

The first hypothesis for image detail restoration states that for a patch in the current decoded frame, there is a high probability that several similar patches with more texture details can be found along its motion trajectory in neighboring frames. To validate this hypothesis, we code several test sequences for three quantization parameter (QP) values, i.e., QP $= 42, 37, 32$. For each 16×16 patch in the decoded frame I_k, we calculate the mean squared error (MSE) between the patch and its uncompressed counterpart, and denote it as MSE_d. Using $I_{k\pm i}$, $i = 1, 2, 3, 4$, as reference frames, we search the best matching patch in all references. Then, the MSE between a matching patch and the original uncom-

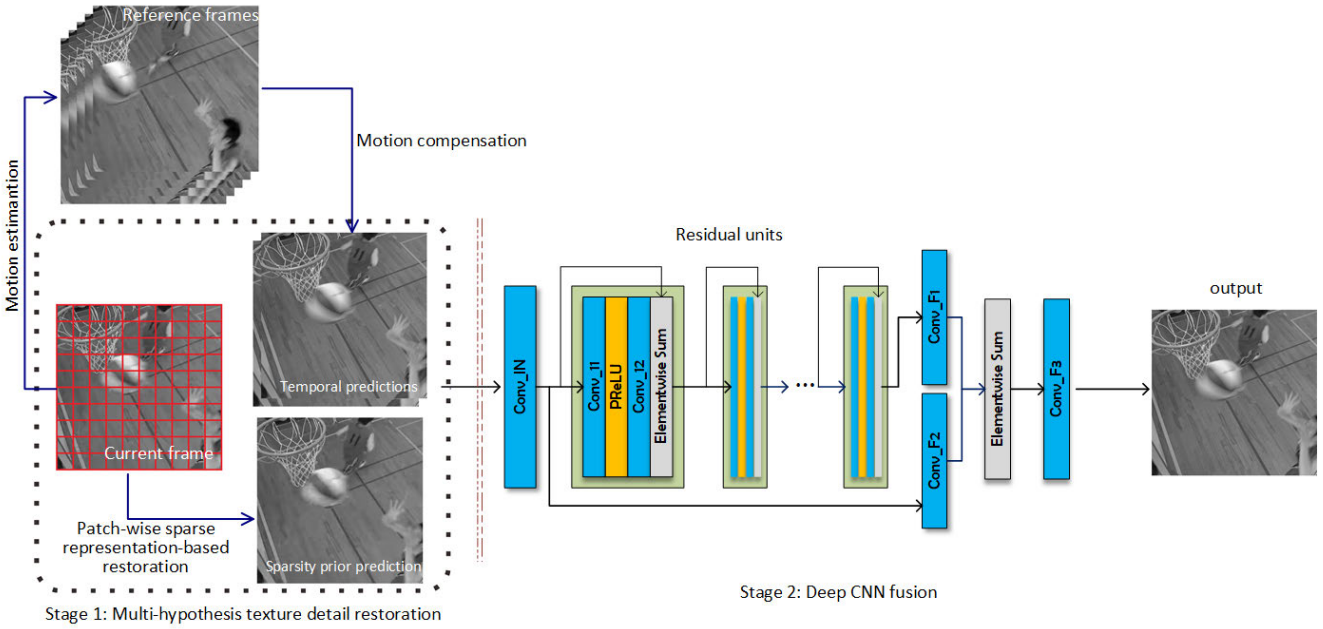

FIGURE 1. Framework of the proposed two-stage method for video compression artifact reduction.

TABLE 1. Percentage values (%) of matching patches.

QP	Sequence	r_1	r_2
42	*BasketballDrive*	81.1	63.2
	BasketballDrill	79.0	60.5
	FourPeople	65.1	50.3
	PartyScene	70.2	47.1
	RaceHorses	78.9	58.8
37	*BasketballDrill*	72.8	48.9
	Cactus	64.3	44.6
	FourPeople	60.5	42.6
	Keiba	64.8	40.2
	Rushhour	78.2	55.7
32	*BQMall*	58.7	35.2
	BQSquare	56.1	37.0
	Cactus	61.1	39.3
	PartyScene	47.6	29.8
	Rushhour	69.9	45.6
	Average	67.2	46.6

pressed one, denoted as MSE_m, is also calculated. For each patch, we count the number of matching patches that are associated with MSE_m smaller than MSE_d. The results for the test sequences are given in Table 1, where r_1 is the percentage of the patches that have at least one matching patch with smaller MSE_m to the number of total patches, and r_2 is the percentage corresponding to the patch having at least two patches with smaller MSE_m.

The results in Table 1 show that on average and over all test sequences, about 67% patches are found to have at least one matching patch with higher quality in neighboring frames, and about 47% are found to have at least two higher quality patches when using 8 reference frames. Understandably, constructing motion-compensated frames by these higher-quality patches and using such frames as input to the deep CNN in the fusion stage could be helpful not only in reducing the training difficulty, but also in improving the visual quality of the predictions of the network.

Note that the original frame (i.e., the reference signal against which the decoded frame is to be compared) is not available at the decoder side, we then confront with a no-reference image quality assessment problem. That is, in the absence of the original uncompressed image, how can we select patches of lower distortions and then using these patches predict motion compensated frames? The strategy in this study uses the default coding setting of HEVC reference software, in which an intra-picture is coded at a relatively high quality. Therefore, the patches located in intra-coded frames are directly accepted as candidates for compensation; for matching patches in inter-coded frames, the approach in [35] is employed to estimate quality scores, and patches with higher scores are selected for compensation. Specifically, assume that we will construct L predictions using temporal correlation hypothesis (we tested $L = 1, 2, 3$ in our experiments). For each patch in the current decoded frame, L patches are selected according to the strategy above, and the ith patch will tile the ith frame in the corresponding position.

B. PATCH PREDICTION USING SPARSITY PRIOR

The second hypothesis utilized in our approach is the sparsity prior, which assumes that most patches in a natural image can be well coded by structural primitives, e.g., edges and textures, and can be represented by a small number of basis functions chosen out of an over-complete code set.

An external database is utilized to learn an one-to-one mapping between HEVC-compressed patches and their corresponding ground-truths. We partition the samples in the database into M clusters by using the k-means clustering algorithm, and hence the samples in a cluster are similar to each other. Then, dictionaries are learned from each of the M clusters. Mathematically, let $P = \{X, Y\}$ denote the training

sets corresponding to a cluster, where X and Y are sets containing quantization degraded and uncompressed image patches, respectively. For each $\sqrt{n} \times \sqrt{n}$ image patch $x_p \in X$ which is cropped from the decoded image at location p, its undegraded counterpart $y_p \in Y$ is extracted from the original uncompressed image at the same position.

Given a cluster P, we aim to learn dictionaries from sets X and Y. The sparse representation (i.e., coding vector α in (1)) for any uncompressed patch in Y is the same as that for its degraded counterpart in X. Following the approach in [36], the learning can be formulated as the following optimization problem:

$$\Phi_d = \arg \min_{\Phi_d,\ \Lambda} \|X - \Phi_d \Lambda\|_2^2 + \lambda \|\Lambda\|_1 \tag{2}$$

and

$$\Phi_o = \arg \min_{\Phi_o,\ \Lambda} \|Y - \Phi_o \Lambda\|_2^2 + \lambda \|\Lambda\|_1 \tag{3}$$

where λ is a parameter that balances the sparsity of the solution and the error term; Φ_o and Φ_d are the dictionaries corresponding to uncompressed and degraded image patches, respectively; $\Lambda = [\alpha_1, \ldots, \alpha_N]$, and each column vector in the matrix satisfies $x_i \approx \Phi_d \alpha_i$ and $y_i \approx \Phi_o \alpha_i$.

Since there always exists fine texture regions in images, learning only one pair of dictionaries is often insufficient to cover all variations of the patches in a cluster. Therefore, in our implementation, we learn multiple dictionary pairs for a cluster, and organize these pairs as a decision tree. Each tree consists of leaf (terminal) and internal (non-terminal) nodes. A leaf node stores a pair of dictionaries which is the optimal solution for estimating the artifact-free patch using a compression degraded patch as input. An internal node serves as a split function which attempts to efficiently select an appropriate dictionary pair for an input sample. For a cluster, the leaf nodes are recursively learned using the corresponding sample set. The maximum depth and the minimum number of samples arriving at a node are chosen to be the stopping criteria, and the growing of a tree will stop if either one is met.

The task for learning an internal node is to find a split function that can correctly partition training samples for each leaf node. There are several choices for this purpose. In this article, we employ Naive Bayes as the weak classifier in internal nodes. Let Φ_o^k and Φ_d^k denote the dictionary pair in the kth leaf node. The class label for (x_i, y_i) is determined by

$$c_i = \begin{cases} 1, & \text{if } \|y_i - \widetilde{y}_i\|_2 < \|y_i - x_i\|_2 \wedge \|y_i - \widetilde{y}_i\|_2 < \delta_T \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

where $\widetilde{y}_i$ is the patch that reconstructed on Φ_o^k with sparse coefficients coded by x_i over Φ_d^k, and δ_T is a threshold. Then, we form the train set as $\{(x_i, c_i)\}_{i=1}^N$, and train a Naive Bayes classifier as the split function for the kth leaf.

The scheme for learning all leaves and the corresponding split functions in a tree is summarized in Algorithm 1.

Algorithm 1 Decision tree learning for a cluster

Input: Data set P for a cluster
Output: The learned decision tree
% Learning dictionaries in leaf nodes;
while *the criteria for stopping the growing is not satisfied* **do**
  Randomly select N sample pairs $\{(x_i, y_i)\}_{i=1}^N$ from P;
  Learn a dictionary pair from these training samples by solving (2) and (3), and store the pair as a leaf;
  $Q \leftarrow \emptyset$;
  for $\forall\, (x_i, y_i) \in P$ **do**
    Estimate the artifact-free patch $\widetilde{y}_i$;
    if $\|y_i - \widetilde{y}_i\|_2 > \|y_i - x_i\|_2$ or $\|y_i - \widetilde{y}_i\|_2 > \delta_T$ **then**
      $Q \leftarrow Q \cup (x_i, y_i)$;
    else
      Skip (x_i, y_i);
    end
  end
  $P \leftarrow Q$;
end
% Learning split functions;
for *each leaf node* **do**
  Assign a class label to each sample in P using (4);
  Form the training set $\{(x_i, c_i)\}$;
  Train a Naive Bayes classifier as the split function;
end

C. VISUAL QUALITY ENHANCEMENT USING A DEEP CONVOLUTIONAL NEURAL NETWORK

The first stage in our method works in a sliding window style. Based on the sparsity prior, we estimate a prediction for each patch in the current decoded frame. The resulted patches are tiled as a reconstructed image. Based on the temporal correlation hypothesis, we search the best matching patch in each reference frame. As a result, we can select several matching patches, and use them to tile motion compensated predictions. The process of the first stage can produce images with more texture details. However, it also creates small artifacts across an image since visible artifacts may occur on block boundaries. To circumvent these artifacts, the second stage in our method proposes a deep CNN that further enhances the visual quality of images.

For restoring a given decoded frame, the degraded image itself, the sparse coding based reconstruction, and several motion compensated frames are stacked as a tensor. The first convolutional layer in the deep CNN (i.e., Conv_IN in Figure 1) takes the tensor as the input to extract feature maps.

At the core of our deep CNN are many stacked residual units which have identical layout. Inspired by [15], [37] for debluring and super-resolution, we do not use batch normalization layers and remove the rectified linear unit which follows the short connection of the original building block

TABLE 2. Typical parameter configurations for the proposed deep CNN.

Layer	Filter	Stride	Channel output	# parameters
Conv_IN	$3 \times 3 \times 3, 24$	1	24	672
8 residual units	$\begin{bmatrix} 3 \times 3 \times 24, & 48 \\ 3 \times 3 \times 48, & 24 \end{bmatrix} \times 8$	1	24	166464
Conv_F1	$3 \times 3 \times 24, 16$	1	16	3472
Conv_F2	$3 \times 3 \times 24, 16$	1	16	3472
Conv_F3	$3 \times 3 \times 16, 1$	1	1	145
Total				174225

as in [38]. As shown in Figure 1, the residual building block in our model uses two convolutional layers. The first layer (e.g., Conv_11) uses $3 \times 3 \times C$ kernels and generates $2C$ feature maps, where C is a constant; whereas the second one (e.g., Conv_12) uses $3 \times 3 \times 2C$ kernels, and generates C feature maps. After the first convolutional layer, a parametric rectified linear unit (PReLU), whose slope for negative inputs is learned from data rather than pre-defined, is used as the activation function. Let z_i and z_{i+1} denote the input and output of the ith residual unit, respectively. The nonlinear mapping of the unit can be formulated as follows [38], [39]:

$$z_{i+1} = z_i + \mathcal{F}(z_i, W_i) \tag{5}$$

where $\mathcal{F}$ represents the residual function, and W_i is the weights of the ith unit.

The reconstruction module in our network includes three convolutional layers (i.e., Conv_F1, Conv_F2 and Conv_F3). As illustrated in Figure 1, there are two inputs for this module, i.e., the input z_1 of the first residual blocks and the output z_{L+1} of the last block. For each input, 16 filters of size $3 \times 3 \times C$ are used to generate feature maps, and then fused via element-wise sum. Finally, the last convolutional layer outputs the desired reconstructed image corresponding to the input tensor using a $3 \times 3 \times 16$ filter.

Assume that the number of the input channels is 3. A typical parameter configuration for the proposed deep CNN is listed in Table 2, where the total number of residual units is chosen to be 8, and the constant $C = 24$.

IV. EXPERIMENTS

A. DATASETS AND TRAINING

For learning the dictionaries to be used for sparse reconstruction and training the deep CNN in our method, we have built a large-scale dataset which was derived from standard test sequences recommended by JCT-VC with four resolutions (Class B, C, D, and E). For each sequence, the corresponding HEVC bitstream is coded by HM 16.0 [40] at three QP values (i.e., 42, 37, 32) using *encoder_lowdelay_P_main.cfg* as the configuration file. For each QP value, a dataset containing several hundred image pairs is constructed. Let it be denoted as $S = \left\{\left(\widetilde{I}_k, I_k\right)\right\}_{k=1}^{N_S}$, where I_k is a frame selected from a test sequence, $\widetilde{I}_k$ is the corresponding decoded frame, and N_s is the total number frame pairs in the dataset. Only the first 30 frames of a sequence are included in the training set; the frames thereafter serve as test set.

For constructing the temporal correlation based predictions, block matching algorithm, which estimates motion on the basis of rectangular blocks and generates one motion vector for each block [41]–[43], is employed to obtain a straightforward and efficient implementation. Though the full search method, which checks all candidate patches to find the best match within a particular window, could finds the best motion vectors in a global sense, its computational requirement is often too high for practical implementation. Therefore, we use an improvement of the successive elimination algorithm (SEA) [41], which excludes many search positions and still allows accuracy comparable to that of the full search. In our experiments, the patch size is fixed to be 16×16, and the search range is determined in relation to the interval between the current frame and the reference frame. To cope with the case of abrupt scene change, when the matching error exceeds a certain threshold, as in [23], the matching patches are discarded and replaced by the patch in the current frame.

For learning decision trees used for sparse coding-based reconstruction, we randomly crop about 500,000 patch pairs from images in S (one from $\widetilde{I}_k$ and the other from I_k at the same position). The size of the patches is set to be 8×8. These patch pairs are then partitioned into clusters, and for each cluster, we learn a decision tree that stores dictionary pairs as the leaf nodes and split functions as internal nodes.

For training our deep CNN, we randomly select a degraded image in the training set, and crop a $B \times B$ patch (we have experimented different patch sizes, e.g., $B = 96, 128, 160$) by randomly selecting the pixel coordinate of the patch. And total of L ($L = 1$, 2 or 3) patches of the same size are cropped at the same position in the motion compensated frames, and one patch is cropped in the sparse coding-based reconstruction. These $L + 1$ patches along with the degraded patch itself are stacked to form a tensor as the input of the network. Then, the patch from the same location in the corresponding uncompressed image serves as the ground-truth. As a result, a training sample is formed, and the flip based data augmentation is also adopted during training.

The proposed network is implemented under the Tensorflow framework and trained using a Nvidia GTX 1080Ti GPU with a mini-batch size of 80 for 128×128 patches. The ADAM optimizer [44] is adopted to optimize the parameters. The learning rate is initialized to be 0.01 and then multiplied by 0.995 after every epoch. The training generally takes about 1000 epoches to converge.

B. RESULTS OF PATCH-BASED RECONSTRUCTION

We now show that the predictions using the temporal correlation and sparsity prior can indeed yield better details and textures, thus allow improvement of artifact reduction in the deep CNN fusion stage. For this purpose, we present part of the results of the predictions using these two hypotheses at QP $= 42, 37$, and show in Figures 2 and 3, respectively. The first column in each figure shows the original frames, the second shows degraded images compressed by HEVC. The predictions derived based on temporal correlation and constructed via the sparse representation are shown in columns 3 and 4, respectively.

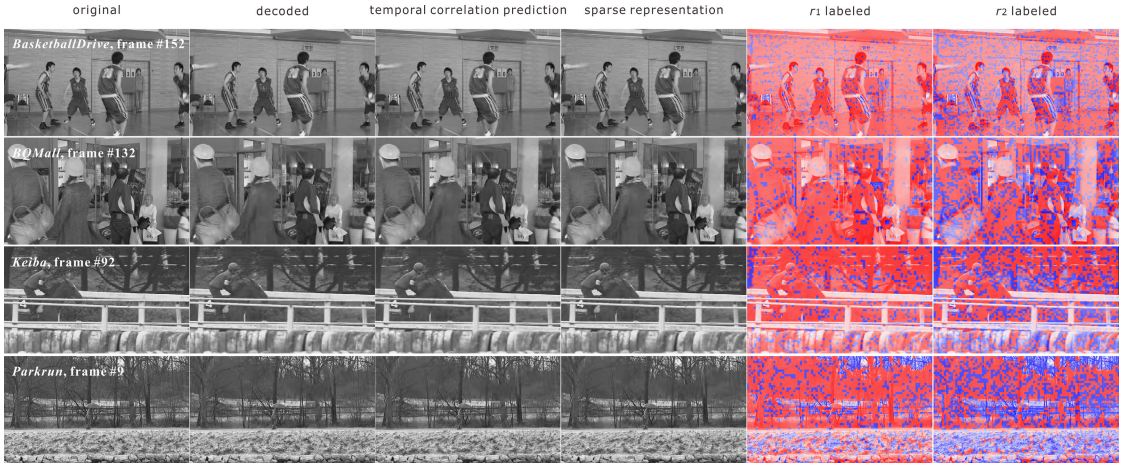

FIGURE 2. Experimental results of temporal correlation and sparsity prior predictions (QP = 42).

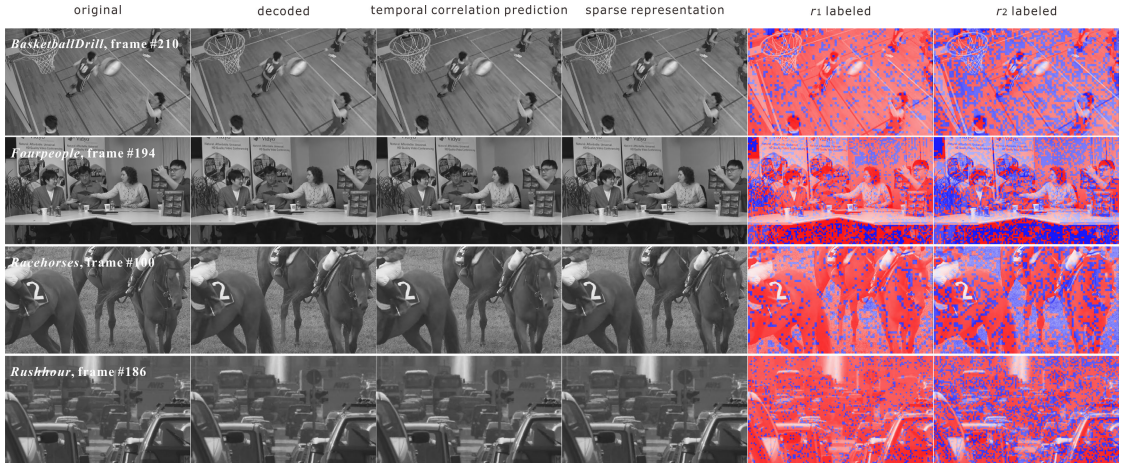

FIGURE 3. Experimental results of temporal correlation and sparsity prior predictions (QP = 37).

From the results, we observe that the predictions (especially the sparse coding-based reconstructions) restore some local texture details. However, some noises are also seemingly introduced. To demonstrate the comparison graphically, we also use MSE as the quality metric. Let x denote a decoded patch, y and $\widetilde{y}$ the uncompressed original and the predicted one corresponding to x, respectively. We calculate and compare two types of MSEs: 1) MSE_d between y and x, and 2) MSE_r between y and $\widetilde{y}$. For each 8×8 patch in the current frame, we count the number of reconstructed patches associated with an MSE_r that is smaller than MSE_d. For the patches that are associated with at least one reconstructed patch with smaller MSE_r, we label them as red regions (other regions shown in blue) and show results in the fifth columns of Figures 2 and 3 (denoted as r_1 labeled). The patches associated with at least two smaller MSE_r predictions are labeled as red regions and shown in the sixth columns (denoted as r_2 labeled) in the two figures.

From both figures, we can see that vast majority of patches have at least one reconstructed patch with higher quality, and about half of them have at least two higher quality predictions. The results confirm that the proposed two hypotheses can predict images with better details.

C. RESULTS OF DEEP CONVOLUTIONAL NEURAL NETWORK

Figures 4 and 5 show the subjective quality performance on some test sequences at QP = 42, 37, respectively. By magnifying the images, we can see that our method can better reconstruct visually pleasing results with sharper edges and more texture details than the other methods under comparison, namely, AR-CNN [19], ARTN [23], and MFQE2.0 [24]. The implementations of these methods are all based on their released source codes, and our codes are also available at *https://github.com/wgchen-gsu/VCAR-CNN*.

The quality enhancement is also objectively evaluated in terms of ΔPSNR and ΔSSIM, which measure the PSNR and SSIM (structural similarity index metric) gain of the enhanced over the original decoded sequences, respectively. Table 3 tabulates ΔPSNR results on some test sequences, with the best results highlighted in boldface type. Table 4 gives average ΔSSIM values of the test sequences.

TABLE 3. Comparison of ΔPSNR (dB) for test sequences at three QP values.

QP	Class	Sequence	Elad *et al.* [11]	AR-CNN [19]	ARTN [23]	MFQE2.0 [24]	Proposed (1M-1S-1D)
42	B	*BasketballDrive*	0.0039	0.0110	0.4332	0.5481	**0.6171**
		Cactus	0.0021	0.0077	0.0328	**0.4011**	0.2415
		Rushhour	0.0030	0.0121	0.7515	0.6159	**0.8017**
		Station2	0.0020	0.0069	0.0034	0.2320	**0.2490**
	C	*BasketballDrill*	0.0033	0.0121	0.6136	0.5740	**0.6414**
		BQMall	0.0030	0.0115	0.4577	**0.5060**	0.3707
		Keiba	0.0032	0.0077	0.6377	0.1062	**0.6514**
		PartyScene	0.0035	0.0064	**0.3661**	0.2904	0.3069
		RaceHorses	0.0022	0.0051	**0.4972**	0.3198	0.3198
	D	*BasketballPass*	0.0030	0.0081	0.4457	**0.5853**	0.4238
		BlowingBubbles	0.0032	0.0065	0.2002	**0.2063**	0.1520
		BQSquare	0.0036	0.0122	0.2122	**0.3345**	0.2408
	E	*FourPeople*	0.0020	0.0062	0.6542	0.6881	**0.7098**
		Parkrun	0.0040	0.0101	0.0150	0.1074	**0.1655**
		Stockholm	0.0030	0.0098	0.0011	0.2339	**0.5353**
		Average	0.0030	0.0089	0.3548	0.3833	**0.4284**
37	B	*BasketballDrive*	0.0022	0.0083	0.3784	0.4650	**0.5198**
		Cactus	0.0018	0.0042	0.3058	**0.5010**	0.3699
		Rushhour	0.0031	0.0121	0.5181	0.5049	**0.5512**
		Station2	0.0016	0.0042	0.0877	0.0921	**0.2763**
	C	*BasketballDrill*	0.0011	0.0037	0.4939	0.5790	**0.5953**
		BQMall	0.0023	0.0067	0.4741	**0.6180**	0.3495
		Keiba	0.0079	0.0191	**0.4565**	0.0939	0.4451
		PartyScene	0.0045	0.0128	0.4488	0.3631	**0.4614**
		RaceHorses	0.0021	0.0062	0.1859	0.3614	**0.3729**
	D	*BasketballPass*	0.0001	0.0017	0.3960	**0.7280**	0.4419
		BlowingBubbles	0.0011	0.0041	0.2772	**0.5330**	0.2395
		BQSquare	0.0020	0.0071	**0.4829**	0.3370	0.4788
	E	*FourPeople*	0.0079	0.0191	0.5730	0.5630	**0.6210**
		Parkrun	0.0061	0.0182	0.0396	0.0110	**0.3066**
		Stockholm	0.0045	0.0148	0.0352	0.0680	**0.2120**
		Average	0.0032	0.0095	0.3435	0.3879	**0.4161**
32	B	*BasketballDrive*	0.0018	0.0059	0.0760	0.1256	**0.1854**
		Cactus	0.0010	0.0072	0.2545	0.2400	**0.3010**
		Rushhour	0.0026	0.0056	**0.4227**	0.2922	0.3176
		Station2	0.0014	0.0042	0.0002	0.0840	**0.1937**
	C	*BasketballDrill*	0.0017	0.0055	0.4985	0.4701	**0.5966**
		BQMall	0.0019	0.0067	0.3564	**0.4692**	0.3574
		Keiba	0.0053	0.0137	**0.3766**	0.1801	0.3023
		PartyScene	0.0031	0.0097	**0.4474**	0.3668	0.4115
		RaceHorses	0.0020	0.005	0.3407	0.3277	**0.3743**
	D	*BasketballPass*	0.0010	0.0028	0.4870	**0.5911**	0.5095
		BlowingBubbles	0.0011	0.0053	0.3119	**0.5141**	0.2714
		BQSquare	0.0016	0.0041	0.4491	0.1898	**0.4673**
	E	*FourPeople*	0.0042	0.0103	0.5681	**0.7073**	0.4926
		Parkrun	0.0032	0.0097	0.0019	0.0448	**0.1836**
		Stockholm	0.0021	0.0088	0.0151	0.0253	**0.1936**
		Average	0.0023	0.0069	0.3071	0.3085	**0.3439**

TABLE 4. Comparison of average ΔSSIM ($\times 10^{-3}$) for test sequences.

QP	Elad *et al.* [11]	AR-CNN	ARTN	MFQE2.0	Proposed
42	0.00	0.03	14.27	13.90	**14.61**
37	0.00	0.01	7.33	7.27	**8.10**
32	0.00	0.00	4.03	3.99	**5.71**

From the results, we can see that the proposed method achieves highly competitive performance compared with other leading compression artifact reduction methods. The performance gain mainly owes to that in the first stage of our method. The sparsity prior and the temporal correlation hypothesis can predict patches with more texture details and therefore can address the challenge of the loss of high frequency contents induced by quantization in video compression.

Following the approach of [45], we evaluate our method using the perception index (PI), which is a no-reference metrics that measures the perceptual quality of a reconstructed image and is calculated as a combination of the no-reference image quality measure Ma_score and the naturalness image quality evaluator (NIQE) value [46]:

$$\mathrm{PI} = \frac{1}{2}\big((10 - \mathrm{Ma\_score}) + \mathrm{NIQE}\big) \tag{6}$$

Recall that a lower PI value indicates better perceptual quality. For jointly quantifying the accuracy (in terms of the root mean square error (RMSE)) and perceptual quality in (6), we compare ARTN, MFQE2.0 and our method on the perception-distortion plane. For every sequence listed in Table 3, we compute the corresponding average PI and RMSE values of all frames of the test sequence and mark a dot on the perception-distortion plane. The results in Figure 6 also demonstrate that our method is admissible among the group of the tested methods.

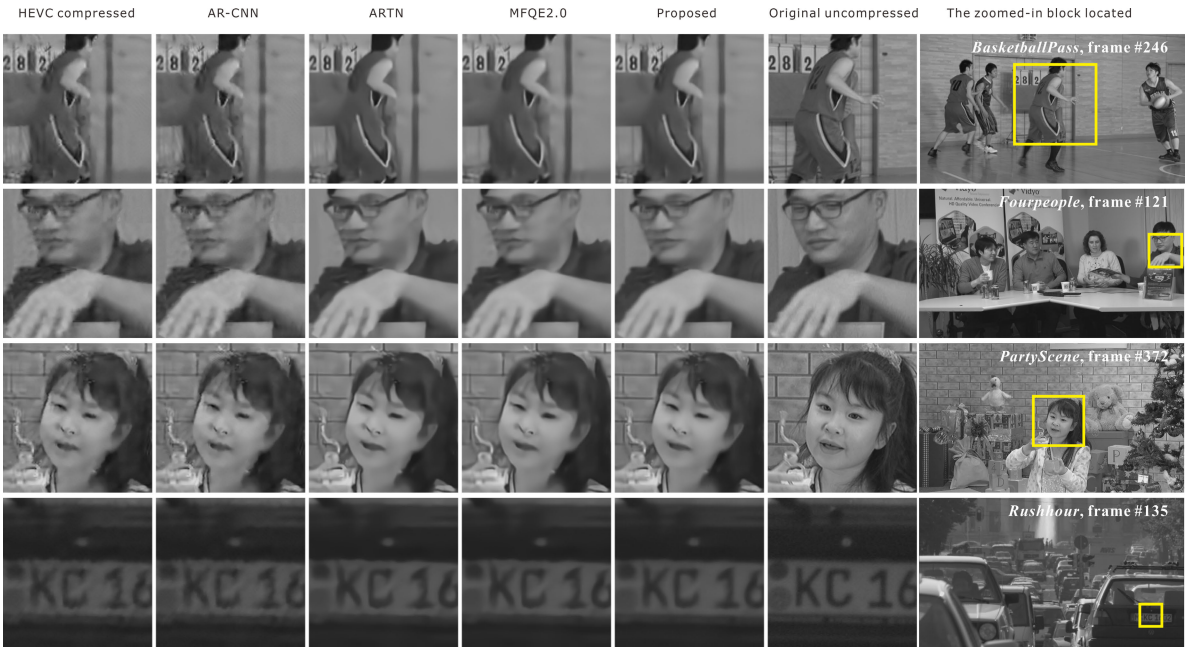

FIGURE 4. Subjective quality comparison on 4 sequences (QP = 42).

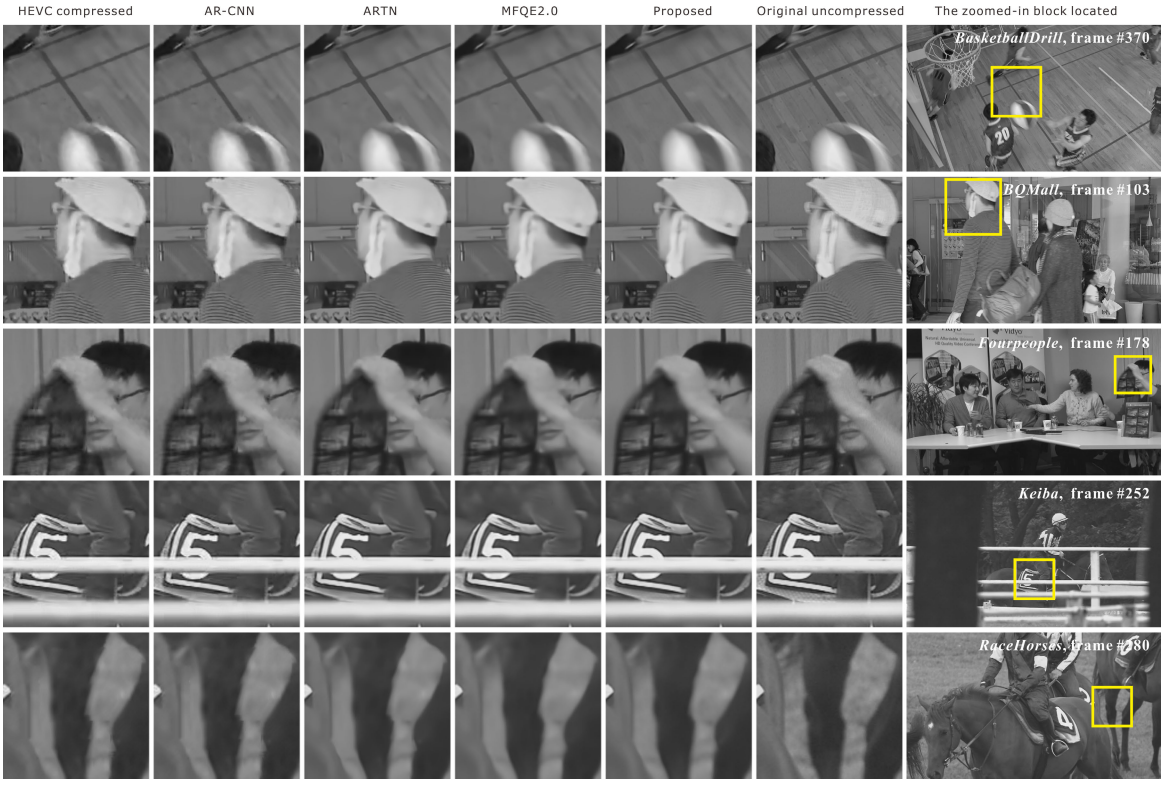

FIGURE 5. Subjective quality comparison on 5 sequences (QP = 37).

In our framework, various combinations of the predictions resulted from temporal correlation and sparsity prior can be used as input. To evaluate the impact of such combinations, we conducted experiments on several different choices. Table 5 tabulates average ΔPSNR and ΔSSIM results for three combinations on the same test sequences that are considered in Table 3, where xM-1S-1D indicates that x motion compensated frames ($x = 1, 2, 3$) and one sparse coding-based prediction, along with the decoded frame itself, are used as the input. For all these combinations, 8 residual units are stacked as the core of our deep CNN model.

We noted that 1M-1S-1D performs the best in most cases. The results suggests that using more motion compensated predictions may not be helpful in enhancing the performance

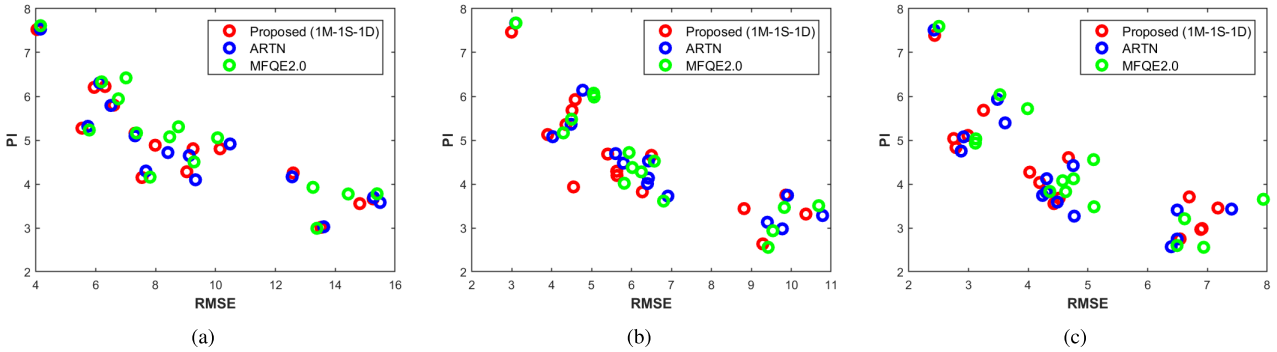

FIGURE 6. Performance comparison on the perception-distortion plane (a) QP = 42; (b) QP = 37; (c) QP = 32.

TABLE 5. Comparison of average ΔPSNR (dB) and ΔSSIM under different combinations of predictions.

QP	1M-1S-1D		2M-1S-1D		3M-1S-1D	
	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM
42	**0.4284**	**0.0146**	0.4183	0.0141	0.3811	0.0132
37	**0.4161**	0.0081	0.3983	**0.0082**	0.3429	0.0078
32	**0.3439**	**0.0057**	0.3257	0.0054	0.3108	0.0051

TABLE 6. Comparison of average ΔPSNR (dB) and ΔSSIM under different number of residual units.

QP	6 residual uints		8 residual units		10 residual units	
	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM	ΔPSNR	ΔSSIM
42	0.4127	0.0139	**0.4284**	**0.0146**	0.4203	0.0141
37	0.3901	0.0080	0.4161	0.0081	**0.4167**	**0.0082**
32	0.3283	0.0053	**0.3439**	**0.0057**	0.3397	0.0055

of the model. The reason behind this might be that as for the temporal correlation based predictions of a given patch, the number of those that have higher quality than the patch is limited, and using more frames formed by tiling these patches of lower quality would not be helpful in enhancing the performance.

We also evaluate the performance while changing the number of residual units in our model. Specifically, we fix the combination of the predictions as 1M-1S-1D and use 6, 8, or 10 residual units in the deep CNN fusion stage. The average ΔPSNR and ΔSSIM values are given in Table 6. The results show that increasing the number of the stacked residual units does not necessarily render improvement of the network performance, possibly because of the difficulty associated with the network training when the network goes deeper.

D. COMPUTATIONAL COMPLEXITY

We evaluate the running time for the proposed method using a computer equipped with a CPU of Intel I9-9900K, 32GB of memory and the GPU of GeForce GTX 1080 Ti. We record the inference time for video sequences at different resolutions. On average, the ARTN [23], MFQE2.0 [24], and our method (1M-1S-1D) takes the computation time of 5.66, 8.12, and 15.07 seconds respectively for processing a 1920 × 1080 (i.e., Class B) frame, and 2.58, 3.55, and 8.17 seconds respectively for processing a 1280 × 720 (i.e., Class E) frame. The running time of the proposed method is about two or three times of that taken by the other methods under comparison. We would like to point out that the first stage of our method (i.e., patch-by-patch detail restoration based on the temporal correlation hypothesis and sparsity prior) is currently implemented in MATLAB and Python without parallelism. Especially, we find in experiments that the sparsity-based processing typically takes about 75% of total computation time for the proposed method. In the future, we intend to optimize the first stage processing with parallel implementation and run optimized codes on a GPU. We believe that the computation time of the proposed two-stage method could be significantly reduced.

V. CONCLUSION

We have proposed in this article a two-stage method to meet the challenge of restoring high frequency contents lost due to quantization for video compression artifact reduction. The first stage aims at image texture detail restoration and performs in a patch-by-patch manner. Based on the sparsity prior, we estimate prediction for every patch in the current decoded frame, and tile the resulted patches as a sparse coding-based reconstruction. Based on the temporal correlation hypothesis, we search the best matching patches in each reference frame, and select a number of matches with more image details to tile motion compensated predictions. In the second stage, we stack a decoded frame and its reconstructed counterparts as a tensor. A deep CNN is proposed to learn the mapping between the input tensor and the original uncompressed image.

The main advantage of the proposed approach is that it typically produces patches with more texture details under the sparsity and the temporal correlation hypothesis in the first stage. Using these high-quality predictions as the input to the deep CNN enables the model to yield the output of high visual quality. Experimental results demonstrate that our proposed method can remarkably improve both the subjective and the objective quality of the compressed video sequences. It should be noted that the multi-hypothesis prediction introduces additional computation. Also note that in the sparse coding-based patch reconstruction, the accuracy and computational efficiency are influenced by the number

of clusters and the number of dictionary pairs for a given cluster. One important problem for future investigation is to determine optimal values for the parameters of the sparse coding-based reconstruction.

REFERENCES

[1] F.-X. Coudoux, M. G. Gazalet, and P. Corlay, "A post-processor for reducing temporal busyness in low-bit-rate video applications," *Signal Process., Image Commun.*, vol. 18, no. 6, pp. 455–463, Jul. 2003.

[2] C.-B. Wu, B.-D. Liu, and J.-F. Yang, "Adaptive postprocessors with DCT-based block classifications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 5, pp. 365–375, May 2003.

[3] S. H. Park and D. S. Kim, "Theory of projection onto the narrow quantization constraint set and its application," *IEEE Trans. Image Process.*, vol. 8, no. 10, pp. 1361–1373, Jan. 1999.

[4] H. Paek, R.-C. Kim, and S.-U. Lee, "A DCT-based spatially adaptive post-processing technique to reduce the blocking artifacts in transform coded images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, no. 1, pp. 36–41, Jan. 2000.

[5] C. Jung, L. Jiao, H. Qi, and T. Sun, "Image deblocking via sparse representation," *Signal Process., Image Commun.*, vol. 27, no. 6, pp. 663–677, Jul. 2012.

[6] J. Zhang, D. Zhao, and W. Gao, "Group-based sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3336–3351, Aug. 2014.

[7] C.-H. Yeh, L.-W. Kang, Y.-W. Chiou, C.-W. Lin, and S.-J. F. Jiang, "Self-learning-based post-processing for image/video deblocking via sparse representation," *J. Vis. Commun. Image Represent.*, vol. 25, no. 5, pp. 891–903, Jul. 2014.

[8] H. Chang, M. K. Ng, and T. Zeng, "Reducing artifacts in JPEG decompression via a learned dictionary," *IEEE Trans. Signal Process.*, vol. 62, no. 3, pp. 718–728, Feb. 2014.

[9] X. Liu, X. Wu, J. Zhou, and D. Zhao, "Data-driven sparsity-based restoration of JPEG-compressed images in dual transform-pixel domain," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 5171–5178.

[10] C. Zhao, J. Zhang, S. Ma, X. Fan, Y. Zhang, and W. Gao, "Reducing image compression artifacts by structural sparse representation and quantization constraint prior," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 10, pp. 2057–2071, Oct. 2017.

[11] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[12] K. Bredies and M. Holler, "A total variation–based JPEG decompression model," *SIAM J. Imag. Sci.*, vol. 5, no. 1, pp. 366–393, Jan. 2012.

[13] T. Meier, K. N. Ngan, and G. Crebbin, "Reduction of blocking artifacts in image and video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 9, no. 3, pp. 490–500, Apr. 1999.

[14] D. Sun and W.-K. Cham, "Postprocessing of low bit-rate block DCT coded images based on a fields of experts prior," *IEEE Trans. Image Process.*, vol. 16, no. 11, pp. 2743–2751, Nov. 2007.

[15] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1132–1140.

[16] Z. Wang, P. Yi, K. Jiang, J. Jiang, Z. Han, T. Lu, and J. Ma, "Multi-memory convolutional neural network for video super-resolution," *IEEE Trans. Image Process.*, vol. 28, no. 5, pp. 2530–2544, May 2019.

[17] K. Jiang, Z. Wang, P. Yi, G. Wang, T. Lu, and J. Jiang, "Edge-enhanced GAN for remote sensing image super-resolution," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5799–5832, Aug. 2019.

[18] 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.

[19] 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)*, Santiago, Chile, Dec. 2015, pp. 1–9.

[20] 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. 1–8.

[21] J. Guo and H. Chao, "One-to-many network for visually pleasing compression artifacts reduction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 4867–4876.

[22] Y. Dai, D. Liu, and F. Wu, "A convolutional neural network approach for post-processing in HEVC intra coding," in *Proc. Int. Conf. Multi. Modeling*, Reykjavik, Iceland, Jan. 2016, pp. 28–39.

[23] J. W. Soh, J. Park, Y. Kim, B. Ahn, H.-S. Lee, Y.-S. Moon, and N. I. Cho, "Reduction of video compression artifacts based on deep temporal networks," *IEEE Access*, vol. 6, pp. 63094–63106, Oct. 2018.

[24] 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.*, early access, Oct. 2, 2019, doi: 10.1109/TPAMI.2019.2944806.

[25] L. Galteri, L. Seidenari, M. Bertini, and A. D. Bimbo, "Deep universal generative adversarial compression artifact removal," *IEEE Trans. Multimedia*, vol. 21, no. 8, pp. 2131–2145, Aug. 2019.

[26] P. Yi, Z. Wang, K. Jiang, Z. Shao, and J. Ma, "Multi-temporal ultra dense memory network for video super-resolution," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 8, pp. 2503–2516, Aug. 2020.

[27] M. A. Robertson and R. L. Stevenson, "DCT quantization noise in compressed images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 1, pp. 27–38, Jan. 2005.

[28] A. Foi, V. Katkovnik, and K. Egiazarian, "Pointwise shape-adaptive DCT for high-quality denoising and deblocking of grayscale and color images," *IEEE Trans. Image Process.*, vol. 16, no. 5, pp. 1395–1411, May 2007.

[29] X. Zhang, R. Xiong, X. Fan, S. Ma, and W. Gao, "Compression artifact reduction by overlapped-block transform coefficient estimation with block similarity," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4613–4626, Dec. 2013.

[30] Y. Dar, A. M. Bruckstein, M. Elad, and R. Giryes, "Postprocessing of compressed images via sequential denoising," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3044–3058, Jul. 2016.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 13–23, 2011.

[32] X. Zhang, R. Xiong, W. Lin, S. Ma, J. Liu, and W. Gao, "Video compression artifact reduction via spatio-temporal multi-hypothesis prediction," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 6048–6061, Dec. 2015.

[33] T. Li, X. He, L. Qing, Q. Teng, and H. Chen, "An iterative framework of cascaded deblocking and superresolution for compressed images," *IEEE Trans. Multimedia*, vol. 20, no. 6, pp. 1305–1320, Jun. 2018.

[34] M. Elad, M. A. T. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proc. IEEE*, vol. 98, no. 6, pp. 972–982, Jun. 2010.

[35] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind image quality assessment: A natural scene statistics approach in the DCT domain," *IEEE Trans. Image Process.*, vol. 21, no. 8, pp. 3339–3352, Aug. 2012.

[36] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. Image Process.*, vol. 19, no. 11, pp. 2861–2873, Nov. 2010.

[37] S. Nah, T. H. Kim, and K. M. Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA, Jul. 2017, pp. 257–265.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.* Amsterdam, The Netherlands, Oct. 2016, pp. 630–645.

[40] *HM 16 Software*. Accessed: Jun. 5, 2019. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/

[41] W.-G. Chen and Y. Ling, "Noise variance adaptive successive elimination algorithm for block motion estimation: Application for video surveillance," *IET Signal Process.*, vol. 1, no. 3, pp. 150–155, Sep. 2007.

[42] H. Yuan, Y. Chang, Z. Lu, and Y. Ma, "Model based motion vector predictor for zoom motion," *IEEE Signal Process. Lett.*, vol. 17, no. 9, pp. 787–790, Sep. 2010.

[43] H. Yuan, J. Liu, J. Sun, H. Liu, and Y. Li, "Affine model based motion compensation prediction for zoom," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1370–1375, Aug. 2012.

[44] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–13.
[45] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor, "The 2018 PIRM challenge on perceptual image super-resolution," 2018, *arXiv:1809.07517*. [Online]. Available: http://arxiv.org/abs/1809.07517
[46] S. Vasu, N. T. Madam, and A. N. Rajagopalan, "Analyzing perception-distortion tradeoff using enhanced perceptual super-resolution network," in *Proc. Eur. Conf. Comput. Vis.*, Munich, Germany, Sep. 2018, pp. 114–131.

WEI-GANG CHEN received the Ph.D. degree from the Department of Computer Science and Technology, Shanghai Jiao Tong University, Shanghai, China, in 2004. He is currently a Professor with the School of Computer Science and Information Engineering, Zhejiang Gongshang University, Hangzhou, China. He has authored or coauthored more than 30 research articles in leading journals and conferences. His current interests include video and image processing, video compression and communication, and object detection.

RUNYI YU (Senior Member, IEEE) received the Ph.D. degree in automatic control from Beijing University of Aeronautics and Astronautics, Beijing, China. He is currently a Professor of electrical and electronic engineering with Eastern Mediterranean University, Famagusta, North Cyprus. He has authored or coauthored more than 60 publications in the general areas of systems and control, and signal processing. His current research interests include sampling theory, tensor analysis, and their applications in data processing.

XUN WANG (Member, IEEE) received the B.Sc. degree in mechanics and the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 1990 and 2006, respectively. He is currently a Professor with the School of Computer Science and Information Engineering, Zhejiang Gongshang University, China. In recent years, he has authored more than 80 articles in journals and conferences. He holds nine authorized invention patents. His current research interests include mobile graphics computing, VR/AR, computer vision, and intelligent information processing. He has five provincial and ministerial level scientific and technological progress awards. He is a member of the ACM and a Distinguished Member of the CCF.