Hindawi
Security and Communication Networks
Volume 2021, Article ID 5568351, 14 pages
https://doi.org/10.1155/2021/5568351

Research Article

A Two-Stage Cascaded Detection Scheme for Double HEVC Compression Based on Temporal Inconsistency

Peisong He,[1] **Hongxia Wang**,[1] **Ruimei Zhang**,[1] **and Yue Li**[2]

[1]*School of Cyber Science and Engineering, Sichuan University, Chengdu 610065, China*
[2]*School of Information Science and Technology, Southwest Jiaotong University, Chengdu 611756, China*

Correspondence should be addressed to Hongxia Wang; hxwang@scu.edu.cn

Received 19 January 2021; Revised 5 March 2021; Accepted 5 April 2021; Published 20 April 2021

Academic Editor: Jinwei Wang

Nowadays, verifying the integrity of digital videos is significant especially for applications about multimedia communication. In video forensics, detection of double compression can be treated as the first step to analyze whether a suspicious video undergoes any tampering operations. In the last decade, numerous detection methods have been proposed to address this issue, but most existing methods design a universal detector which is hard to handle various recompression settings efficiently. In this work, we found that the statistics of different Coding Unit (CU) types have dissimilar properties when original videos are recompressed by the increased and decreased bit rates. It motivates us to propose a two-stage cascaded detection scheme for double HEVC compression based on temporal inconsistency to overcome limitations of existing methods. For a given video, CU information maps are extracted from each short-time video clip using our proposed value mapping strategy. In the first detection stage, a compact feature is extracted based on the distribution of different CU types and Kullback–Leibler divergence between temporally adjacent frames. This detection feature is fed into the Support Vector Machine classifier to identify abnormal frames with the increased bit rate. In the second stage, a shallow convolutional neural network equipped with dense connections is designed carefully to learn robust spatiotemporal representations, which can identify abnormal frames with the decreased bit rate whose forensic traces are less detectable. In experiments, the proposed method can achieve more promising detection accuracy compared with several state-of-the-art methods under various coding parameter settings, especially when the original video is recompressed with a low quality (e.g., more than 8%).

1. Introduction

With the rapid development of communication network (e.g., 5th generation mobile networks, 5G [1]) and video compression technologies (e.g., High Efficiency Video Coding [2]), digital video has become one of the most ubiquitous methods to access latest news. However, using sophisticated edition tools, it is very easy to tamper the contents in digital videos by forgers, which has posed a great threat to the authenticity and integrity of digital videos transmitted over the communication networks. In video forensics, detection of double compression can be regarded as the first step to analyze whether a suspicious video undergoes the tampering operation. The reason behind this is that to generate a tampered video, the forger needs to decompress the original video into a frame sequence and conduct intra-frame or inter-frame tampering operations to modify some specific video contents. Then, the tampered frame sequence is re-encoded as a video file. Therefore, the detection of double compression has drawn attention of researchers in the field of multimedia forensics.

In the last decade, numerous detection methods of double compression have been proposed successfully. Existing detection methods can be divided into two categories according to whether the structures of Group of Picture (referred to as GOP) in the original video and its recompressed version are aligned or not. For detection of double compression with the aligned GOP structure, the most primary clue is the abnormal statistics of double compressed I-frame (namely, Intra-coded frame). Some hand-crafted features are designed using first digit law [3] and Markov statistics [4] of quantized DCT (Discrete Cosine

Transform) coefficients. Then, these features are combined with a traditional classifier (e.g., Support Vector Machines, referred to as SVM) to detect double compression. On the other hand, for detection of double compression with the mismatched GOP structure, the abnormal variation of coding information in relocated I-frames is the most significant forensic traces, where relocated I-frames denote recompressed P-frames (namely, Inter-coded frames) which were I-frames in the original video. The researchers proposed different types of measurement sequences, including prediction residuals [5, 6], variation of macroblock prediction footprint [7], and block artifact [8–10], to expose the periodic occurrence of relocated I-frames. More recently, deep-learning-based methods [11, 12] are applied to locate relocated I-frames based on convolutional and recurrent neural network [13].

In practical applications, videos are likely to be recompressed with various bit rates, which may be larger or smaller than the original bit rate on different degrees. For example, videos transmitted over the communication network are always recompressed with a decreased bit rate to meet the bandwidth constraint [14]. On the other hand, forgers may re-encode video clips with the increased bit rate before splicing video clips with different bit rates [15]. However, most existing methods proposed a universal detector for different settings of recompression bit rates, which are hard to provide reliable detection results when the variation of bit rates is dramatic. To overcome the aforementioned limitations, we propose a two-stage cascaded detection scheme for frame-wise detection of double HEVC compression based on temporal inconsistency, where frame-wise detection means determining whether relocated I-frames exist in a suspicious video. In this work, we first analyzed statistics of coding information in HEVC videos, such as block size and prediction mode of Coding Unit (referred to as CU), and found that CU types in relocated I-frames have quite dissimilar properties between videos recompressed with the increased bit rate and the decreased bit rate. It motivates us that it should be more suitable to detect relocated I-frames in the aforementioned two cases separately. In the proposed scheme, a short-time video clip which contains continuous three frames is treated as an input sample. Two attributes of CU are considered, including block size and prediction mode, to construct CU information map using our proposed value mapping strategy. In the first stage, a compact feature is designed based on the distribution of different CU types and their temporal inconsistency measured by Kullback–Leibler divergence and combined with the SVM classifier to detect relocated I-frame with the increased bit rate (referred to as TypeI P-frames). In the second stage, to explore more slight traces of relocated I-frames with the decreased bit rate (referred to as TypeII P-frames), we proposed a shallow convolution neural network (CNN) equipped with dense connections, which can jointly learn robust spatiotemporal deep representations in compression domain. The main contributions of the proposed method are summarized as follows:

(i) To achieve more robust detection capability of double HEVC compression with various recompression bit rates, a two-stage cascaded detection scheme is proposed based on temporal inconsistency of CU information map. It is different from most existing methods which only constructed a universal detector.

(ii) In the first stage, a compact feature is designed leveraging the distribution of different CU types (considering block size and prediction mode) and their K-L divergence to describe the temporal inconsistency. Then, this feature is fed into the trained SVM classifier to obtain results of relocated I-frames with the increased bit rate (TypeI P-frames).

(iii) In the second stage, a shallow CNN equipped with dense connections is constructed, which can jointly learn spatiotemporal deep representations from both low-level patterns and high-level forensic semantics by feature reuse for relocated I-frames with the decreased bit rate (TypeII P-frames).

(iv) Extensive experiments have been conducted, which considered various coding parameter settings, such as different bit rates, GOP sizes, transcoding processes, and so on. Experimental results verified the more reliable and robust detection capability of the proposed detection scheme compared with several state-of-the-art methods.

2. Related Works

In this work, we focus on the frame-wise detection of double HEVC compression with the mismatched GOP, since HEVC is one of the most advanced video coding standards [2], and double compression with the mismatched GOP is more likely to occur in realistic forensic scenarios [16]. According to the feature extraction process, existing methods can be divided into two categories, including hand-crafted feature-based methods and deep-learning-based methods.

2.1. Hand-Crafted Feature-Based Methods. Relocated I-frame is the most significant clue to detect double compression with the mismatched GOP structure. In early studies, researchers found that relocated I-frames can cause abnormal increment of prediction residuals distinctly and applied the prediction residual sequence [5, 6] and its modified version to conduct detection. Except for prediction residuals, relocated I-frames can cause other coding information to perform abnormal variations. For example, Vazquez-Padin et al. [7] leveraged the variation of macroblock prediction footprints (referred to as VPF) to measure the periodic occurrence of relocated I-frames. This method achieved promising detection performance with different coding parameter settings. In [17], the same authors extended the extraction process of VPF by involving motion vectors to obtain a new feature, called as generalized VPF, which can be used to formulate the measurement sequence or construct the threshold-based classifier to identify relocated I-frames. For HEVC videos, Jiang et al. [18] applied the low-order statistics of Prediction Unit (PU) types in a GOP unit as the feature to detect double compression. On the other hand, traces of double compression left in decompression (pixel) domain can also be used to

construct the measurement sequence to expose relocated I-frames, such as block artifacts [10] and blurring artifacts [19]. However, traces in decompression domain are easier to be degraded by the severe lossy quantization in the re-encoding process. More recently, for HEVC videos, artifacts in both compression domain (e.g., PU types) and decompression domain (optical flow) are combined to expose relocated I-frames [20].

2.2. Deep-Learning-Based Methods. Deep learning, especially convolutional neural network and recurrent neural network, has been applied in the field of computer vision successfully. Due to the strong learning capability of hierarchical representations, researchers also applied deep learning to conduct frame-wise detection of double compression. In [11], He et al. proposed a frame-wise detection method based on CNN, where decompressed frames are stacked together as the input sample. This network initialized with a preprocessing layer can extract the high-frequency components of input sample. Global average pooling and 1×1 convolutional kernel were considered in the network architecture to mitigate the influence of overfitting. Based on the network in [11], Nam et al. [12] proposed a two-stream CNN which can incorporate both decompressed I-frames and P-frames in a GOP unit to detect double compression. However, this method can only provide GOP-wise detection results instead of locating relocated I-frames. Different from [11, 12], the authors in [21] applied the coding information in compression domain as the input and designed a hybrid network architecture which combined CNN and Long Short-Term Memory (LSTM) to learn spatiotemporal representations. Experiments demonstrated that this method can achieve more robust detection capability when original videos are recompressed with a low quality.

3. Preliminaries

In this section, the generation process of double compressed videos is first introduced. Then, the statistics of CU types are analyzed for different kinds of P-frames.

3.1. The Generation Process of Single and Double Compressed Videos. In this section, the generation process of single and double compressed videos is briefly introduced. For a given raw video sequence ($\mathbf{F} = \{F_1, F_2, \ldots, F_T\}$, where F_T denotes the tth raw frame and T denotes the total number of frames), it is encoded with the bit rate B_1 and the GOP size G_1 to obtain the single compressed video (V_s). For simplicity, B-frames (Bidirectional predicted frames) are not considered in this work. As shown in Figure 1, the intra-coding and inter-coding processes in single compression can be formulated as follows:

$$\begin{aligned} P_t^{(1)} &= I\left(F_t\right), \\ P_{t-1}^{(1)} &= F_{t-1} - M\left(F_{t-1}, \widehat{F}_{t-2}^{(1)}\right), \end{aligned} \tag{1}$$

where $I(\cdot)$ denotes the intra-prediction process; $M(\cdot,\cdot)$ denotes the motion prediction; $P_t^{(s)}$ denotes the (intra or inter) prediction residual of tth frame in the sth compression; and $\widehat{F}_t^{(s)}$ denotes the tth decompressed frame after the sth compression. In the inter-coding process of F_{t-1} as shown in equation (1), the magnitude of prediction residuals depends on two factors, including the temporal variation of video contents and the error propagation caused by motion compensation within a GOP unit. In the HEVC standard, Coding Unit (CU) can be regarded as the basic unit to define a region using the same prediction mode (intra-coded or inter-coded), which is organized in a coding tree unit. Different from a fixed 16 × 16 macroblock used in MPEG-2/4 and H.264/AVC [22], HEVC standard allows the more flexible block partition of CUs to achieve better compression efficiency. More specifically, for static regions and contents with the smooth motion, the HEVC encoder is more likely to adopt inter-coded CU (P-CU) or skipped CU (S-CU) with a large block size. The P-CU applies the motion compensation to reduce temporal redundancy, and S-CU can be regarded as a special type of P-CU whose motion vector differences and prediction residuals are zero. On the other hand, for fast moving object with deformation, the HEVC encoder prefers to choose intra-coded CU (I-CU) with the small block size to achieve better balance between compression efficiency and video quality. I-CU can conduct spatial prediction with the neighboring reconstructed pixels.

Then, V_s is decompressed as the frame sequence and re-encoded with the bit rate B_2 and the GOP size G_2 ($G_1 \neq G_2$) to obtain the double compressed video (V_d) as shown in Figure 1. The inter-coding process in recompression can be formulated as follows:

$$\begin{aligned} P_t^{(2)} &= \widehat{F}_t^{(1)} - M\left(\widehat{F}_t^{(1)}, \widehat{F}_{t-1}^{(2)}\right), \\ P_{t-1}^{(2)} &= \widehat{F}_{t-1}^{(1)} - M\left(\widehat{F}_{t-1}^{(1)}, \widehat{F}_{t-2}^{(2)}\right). \end{aligned} \tag{2}$$

It can be observed in Figure 1 that there are two kinds of recompressed P-frames in double compressed videos, namely, relocated I-frames (e.g., the tth frame) and P-P frames (e.g., the $(t-1)$th frame), where P-P frames are re-encoded P-frames which were P-frames in the original video. It has been widely studied in previous works [10, 17] that the magnitude of prediction residuals in relocated I-frames has abnormal increment due to the weak correlation between the current P-frame ($\widehat{F}_t^{(1)}$) and its reference frame ($\widehat{F}_{t-1}^{(1)}$) in the re-encoding process. This weak correlation is due to that tth frame and $(t-1)$th frame located in different GOP units, which were encoded by intra-prediction $I(\cdot)$ and inter-prediction $M(\cdot,\cdot)$ processes, respectively, as shown in equation (1). Consequently, the error propagation in a GOP unit is unrelated to its next GOP unit.

Although the occurrence of relocated I-frames is caused by the mismatched GOP structure, the statistics of coding information depend on the specific recompression bit rate adopted in the re-encoding process. Existing methods rarely consider the unique properties of relocated I-frames with different recompression bit rates. In the next section, the statistics of CU information on different recompression scenarios will be analyzed.

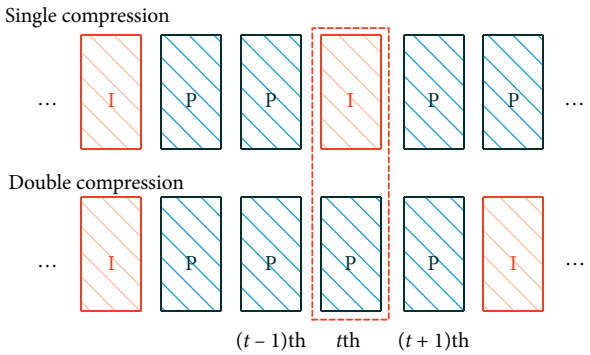

FIGURE 1: The generation process of single and double compressed videos.

3.2. Statistical Analysis of CU Information with Different Recompression Bit Rates. Bit rate is the primary factor to control the quality of compressed videos, especially for network transmission with the limited bandwidth. In this section, we analyze how statistics of CU information perform when a single compressed video is recompressed with different bit rates. We consider two types of recompression processes, including recompression with the increased bit rate and recompression with the decreased bit rate. For the first case, double compressed videos are also called as fake bit rate videos in previous works [23, 24], where the video quality of fake bit rate videos cannot be improved compared with their original videos. On the other hand, for the second case, forensic traces of recompression, such as relocated I-frames [10, 17], suffer from a more distinct degradation caused by the severe lossy quantization. Note that we do not consider recompressed videos with the same coding parameters, such as bit rate, due to the following reasons. (1) The default setting in video editing tools is different from the capture device in most cases. (2) The potential transcoding process during the network communication is uncontrollable. Besides, for this special case, some existing methods can be used to conduct the supplementary analysis [25].

To analyze the statistics of coding information, we calculate the ratio of different CU types in each P-frame and display their distributions in different categories of P-frames using boxplot. In this work, the block size and prediction mode are considered as two attributes of each CU. Consequently, there are 12 CU types, since there are four kinds of block size ({8, 16, 32, 64} with the default setting in the main profile of HEVC standard), and there are three kinds of prediction modes, including I-CU, P-CU, and S-CU. We consider three categories of P-frames, including relocated I-frames with the increased bit rate (TypeI P-frames), relocated I-frames with the decreased bit rate (TypeII P-frames), and other P-frames which contain P-P frames and single compressed P-frames (referred to as TypeIII P-frames). For a specific CU type, its ratio in a P-frame is calculated as follows: $r = N_p/N_t$, where N_p denotes the number of 4×4 sub-blocks belonging to this CU type in this P-frame and N_t denotes the total number of 4×4 sub-blocks in this P-frame. Training samples in Section 5.1 are adopted to calculate the ratios of different CU types. We adopted boxplot to display the distributions of CU types' ratios in different categories of P-frames. For the ratios of a specific CU type from one category of P-frame, we need to calculate their upper margin, upper quartile, median, lower quartile, and lower margin to draw boxplot [26], where outliers whose values are larger than upper margin or lower than lower margin are marked as red cross as shown in Figure 2. We can draw the following conclusions based on the boxplots of different CU types' ratios in different categories of P-frames:

(i) TypeI P-frames contain a higher ratio of I-CU on average compared with other two categories of P-frames for all kinds of block sizes, except that 64×64 I-CU rarely occurs in all categories of P-frames. It implies that temporal inconsistency caused by the mismatched GOP can increase the difference between adjacent two frames which makes the encoder prefer to apply intra-prediction modes in TypeI P-frames.

(ii) TypeI P-frames contain a lower ratio of S-CU on average compared with other two categories of P-frames especially for the relatively large block size, such as 64×64 and 32×32.

(iii) For P-CU, encoders prefer to select CUs with smaller block sizes (e.g., 8×8 and 16×16) in TypeI P-frames due to the abnormal increment of prediction residuals as claimed in Section 3.1.

(iv) Although mean values of CU types' ratios between TypeI P-frames and other two categories of P-frames are discriminate for I-CU and S-CU, the dynamic range of each CU type's ratios is very large, which infers that the statistics of CU types may be influenced by video contents.

(v) It is hard to discriminate between the statistics of TypeII P-frames and TypeIII P-frames by only leveraging CU types' ratios.

Conclusions 1 to 4 infer that it is possible to design compact features to detect TypeI P-frames based on the ratios of different CU types. Conclusion 5 illustrates that only applying low-order statistics (different CU types' ratios) is insufficient to expose TypeII P-frame. Therefore, more discriminative patterns in both spatial and temporal domain should be leveraged to learn robust representations. In the next section, we will propose a two-stage cascaded detection scheme to identify relocated I-frames in double HEVC compression.

4. The Proposed Method

As mentioned in Section 3.2, relocated I-frames perform dissimilar statistics of CU types. It is hard to reveal relocated I-frames with various bit rates by only leveraging one universal detector. In this work, we proposed a two-stage cascaded detection scheme of double HEVC compression based on temporal inconsistency as shown in Figure 3, which aims to provide robust detection capability for recompression with different bit rates. To describe the temporal inconsistency of relocated I-frames, we first construct the CU information map of each frame. Then, a

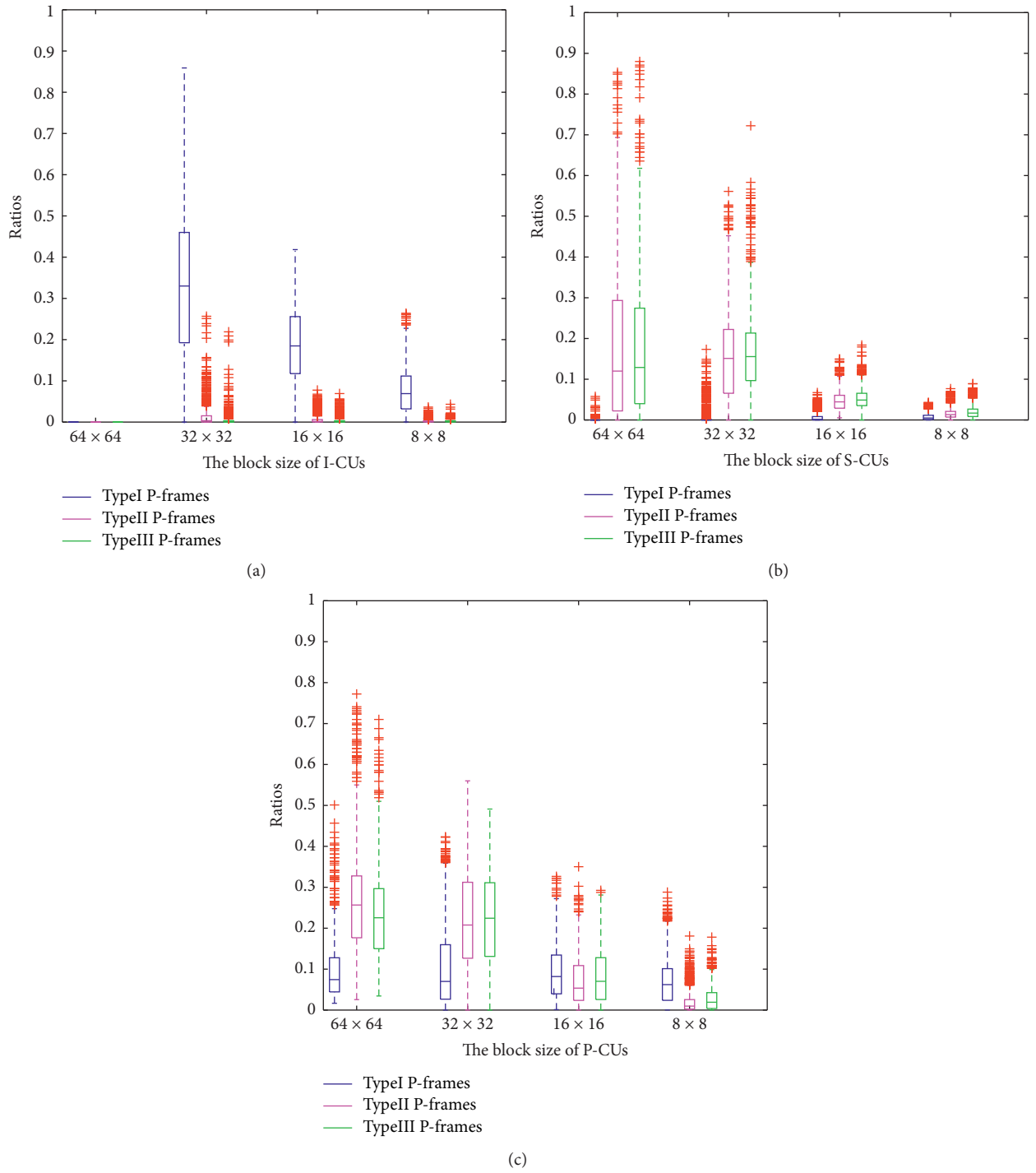

Figure 2: The distributions of CU types' ratios in different P-frames. (a) The distributions of I-CU's ratios with different block sizes. (b) The distributions of S-CU's ratios with different block sizes. (c) The distributions of P-CU's ratios with different block sizes.

compact and efficient detection feature is extracted to distinguish TypeI P-frames and other categories based on the distributions of CU types and its temporal inconsistency measured by Kullback–Leibler divergence (referred to as K-L divergence). To further classify TypeII and TypeIII P-frames, a shallow convolutional neural network equipped with dense connections is constructed to jointly learn spatiotemporal deep representations of both low-level patterns and high-level forensic semantics from CU information maps. Finally, we can obtain the detection results of relocated I-frames.

4.1. Constructing the CU Information Map. For a given frame (F_t, where $t \in \{1, \ldots, T_f\}$ and T_f denotes the total number of P-frames in a video), we first extract CU's information of

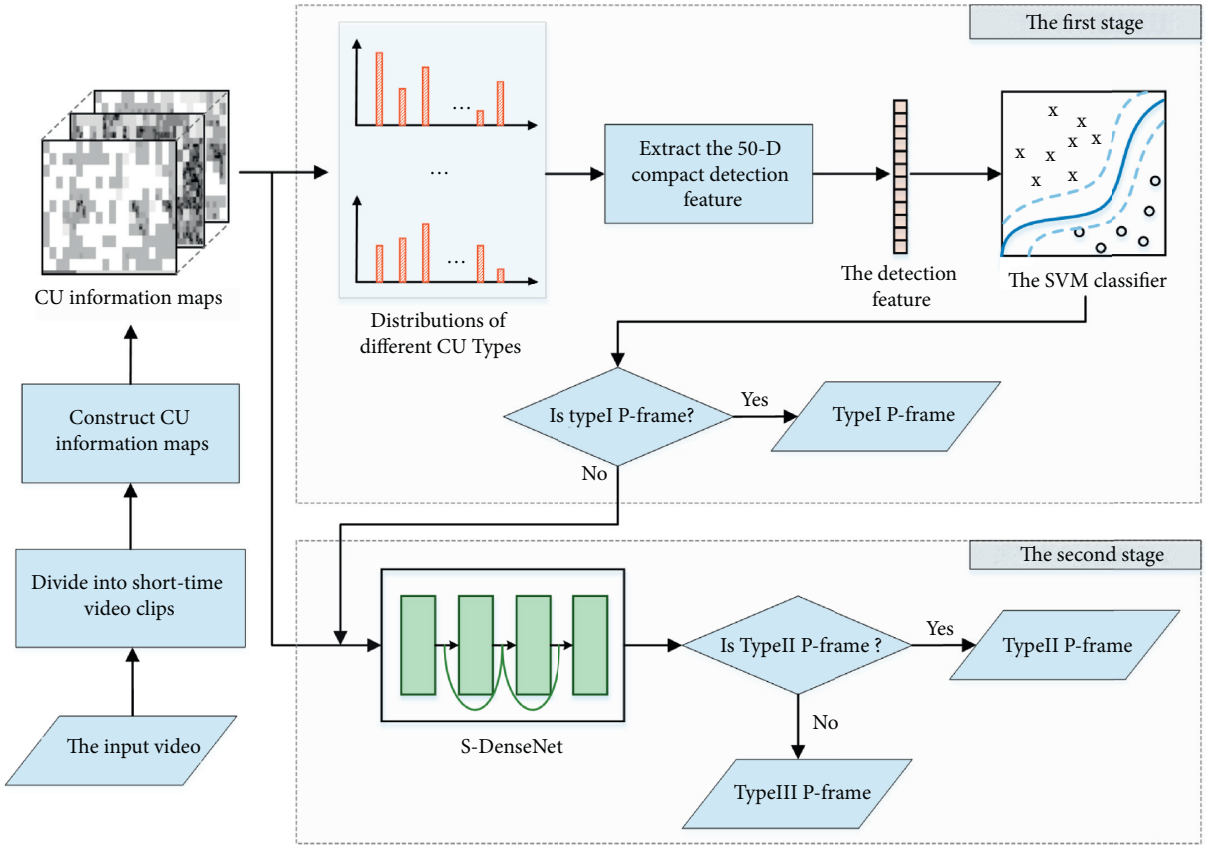

Figure 3: The framework of the proposed scheme.

the current frame (F_t) and its adjacent two frames, namely, F_{t-1} and F_{t+1}. In this work, we consider the block size, prediction mode, and motion vector of each CU to construct the CU information map ($\mathbf{C}_t$) of tth frame. For each 4×4 sub-block with the spatial index (i, j) in the tth frame ($i \in \{1, 2, \ldots, \lfloor M/4 \rfloor\}$ and $j \in \{1, 2, \ldots, \lfloor N/4 \rfloor\}$, where $M \times N$ is the spatial resolution of input video), its value in $\mathbf{C}_t$ is calculated as follows:

$$\mathbf{C}_t(i, j) = \frac{1}{N_c}\left[4 \cdot \mathbf{D}_t(i, j) + \mathbf{P}_t(i, j)\right], \tag{3}$$

where $\mathbf{D}_t(i, j)$ denotes the mapped value depending on the block size of the CU which the (i, j) sub-block belongs to. Specifically, block sizes $\{8 \times 8, 16 \times 16, 32 \times 32, 64 \times 64\}$ are mapped to the values $\{3, 2, 1, 0\}$, respectively; $\mathbf{P}_t(i, j)$ denotes the mapped value depending on the prediction mode of the CU which the (i, j) sub-block belongs to. Similar to previous works [21], motion vectors are also considered to provide useful forensic clues. Depending on whether motion vectors of P-CUs are zero or not, P-CUs can be further divided into two categories, including P-CU with non-zero motion vectors (non-ZMV P-CU) and P-CU with zero motion vectors (ZMV P-CU). Then, prediction modes {I-CU, non-ZMV P-CU, ZMV P-CU, and S-CU} are mapped to the values $\{3, 2, 1, 0\}$, respectively. In our value mapping strategy, the higher value of $\mathbf{C}_t(i, j)$ presents the stronger temporal inconsistency in local regions between adjacent frames. Consequently, there are $N_c (= 4 \times 4 = 16)$ kinds of CU types in the proposed method. Obviously, for a given frame F_t, elements in its CU information map $\mathbf{C}_t \in \mathbb{R}^{\lfloor M/4 \rfloor \times \lfloor N/4 \rfloor}$ are within the range of $[0, 1]$.

4.2. Stage 1: Detection with the Compact Feature Based on the Distribution of CU Types. In the first stage, we design a compact feature to discriminate TypeI P-frames and other two categories, since the temporal inconsistency can be described by different CU types' ratios based on the analysis in Section 3.2. To capture the temporal variation between the suspicious frame and its adjacent frames, we conduct the following steps to extract the detection feature:

(i) For the tth frame, we calculate the distribution of different CU types ($\mathbf{h}_t$) as follows:

$$\mathbf{h}_t(k) = \frac{1}{M' \times N'} \sum_{i=1}^{M'} \sum_{j=1}^{N'} \begin{cases} 1, & \text{if } \mathbf{C}_t(i, j) = \dfrac{k}{N_c}, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

where k denotes the type of CU which (i, j) 4×4 sub-block belongs to, $k \in \{0, \ldots, N_c - 1\}$, and $M' \times N' = \lfloor M/4 \rfloor \times \lfloor N/4 \rfloor$. The value of $\mathbf{h}_t(k)$ denotes the ratio of k-th CU type in the tth frame.

(ii) The temporal inconsistency between the current frame (F_t) and its adjacent frames (F_{t-1} and F_{t+1}) is

measured by K-L divergence, where K-L divergence is a widely used measurement of similarity between two discrete distributions. It can be formulated as follows:

$$s_1 = D_{\mathrm{KL}}\left(\mathbf{h}_{t-1} \| \mathbf{h}_t\right), \quad s_2 = D_{\mathrm{KL}}\left(\mathbf{h}_{t-1} \| \mathbf{h}_t\right), \tag{5}$$

where $D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q})$ denotes the function to obtain the K-L divergence between the distribution $\mathbf{p}$ and $\mathbf{q}$. More specifically, $D_{\mathrm{KL}}(\mathbf{p}\|\mathbf{q}) = \sum_{l=1}^{L} \mathbf{p}(x_l)\log((\mathbf{p}(x_l)/\mathbf{q}(x_l) + \varepsilon) + \varepsilon)$ where L denotes the total number of elements in the distribution $\mathbf{p}$ and $\mathbf{q}$; ε denotes a very small value (e.g., 10^{-15}) which is used to avoid the indeterminate results caused by the potential zero elements in the distribution $\mathbf{p}$ or $\mathbf{q}$.

(iii) Finally, the detection feature $\mathbf{f}_t$ is constructed by concatenating $\mathbf{h}_{t-1}, \mathbf{h}_t, \mathbf{h}_{t+1} \in \mathbb{R}^{1\times 16}$ in equation (4) and $s_1, s_2 \in \mathbb{R}$ in equation (5), which is formulated as

$$\mathbf{f}_t = \left[\mathbf{h}_{t-1}, \mathbf{h}_t, \mathbf{h}_{t+1}, s_1, s_2\right] \in \mathbb{R}^{1\times 50}. \tag{6}$$

Then, the detection feature ($\mathbf{f}_t$) of the tth frame is fed into a trained SVM classifier applying RBF (radial basis function) kernel to obtain detection results of TypeI P-frames (relocated I-frames with the increased bit rate).

4.3. Stage 2: Detection with the Shallow CNN Equipped with Dense Connections. In the case where relocated I-frames are recompressed with the decreased bit rate, forensic traces of recompression operation are less detectable due to the degradation caused by the severe lossy quantization. In other words, the difference between TypeII P-frames (relocated I-frames with the decreased bit rate) and TypeIII P-frames (single compressed P-frames and P-P frames) is much slighter, which is necessary to explore discriminative clues in both spatial and temporal domains. In this work, we proposed a shallow CNN equipped with dense connections (S-DenseNet) to learn spatiotemporal representations of double HEVC compression, where dense connections have been applied to deal with the image classification task successfully [27]. CU information maps of continuous three frames are constructed and then resized into $N_b \times N_b$ in spatial domain considering both detection performance and computational cost. The preprocessed CU information maps are used as the input sample (namely, $\{\mathbf{C}_{t-1}', \mathbf{C}_t'\mathbf{C}_{t+1}'\} \in \mathbb{R}^{3\times N_b\times N_b}$, and $\mathbf{C}_t'$ denotes the preprocessed CU information map of the tth frame).

Figure 4 presents the network architecture of our proposed S-DenseNet. It includes six convolutional modules (referred to as conv modules) and a transition module, where each conv module consists of a convolutional layer, a batch normalization layer, and a ReLU layer, as shown in Figure 5. The transition module aims to improve the compactness of the network, which has the identical structure to conv module, except that there is a average pooling layer (2×2 pooling operation window with the stride 2×2) following the ReLU layer to reduce the spatial size of output feature maps. The detailed setting of the convolutional kernel in transition module is [96, 96, 1×1, 1×1]. Besides, in Figure 4, "FC" denotes a fully connected layer with 128 neurons and "softmax" denotes a fully connected layer with 2 neurons followed by a softmax layer. The cross-entropy loss is applied to optimize the network weights, and we do not apply any other regularization terms in loss function.

It can be observed that, for each conv module, the output feature maps of several preceding conv modules are concatenated channel-wise and used as the input of the next conv module. Applying dense connections in network architecture has the following advantages: (1) it is useful to mitigate the vanishing-gradient problem during the optimization process of network parameters, especially when the potential values in CU information maps are limited, namely, $N_c = 16$ in our method. (2) Dense connections support the feature reuse which can help the network learn spatiotemporal representations from both low-level patterns in early layers and high-level forensic semantics in top layers.

After an input sample being fed into the trained shallow DenseNet, the output vector $[p_0, p_1]$ from the softmax layer can be obtained, where p_0 and p_1 present the probabilities that the input sample belongs to TypeII P-frame and TypeIII P-frame, respectively. Then, p_0 is applied as the detection score. If $p_0 > T_p$, the input sample is classified as TypeII P-frame where T_p is set as 0.5. Otherwise, the input sample is classified as TypeIII P-frame.

5. Experiments

In this section, several experiments are conducted to evaluate the detection performance in various scenarios, such as different bit rates, different GOP sizes, and so on.

5.1. Database. To construct the database of single and double compressed HEVC videos, 26 raw videos (YUV sequences) are collected from the Internet, which have various video contents. The list of raw videos and the downloading address are presented in Appendix. Several examples are shown in Figure 6. To align different resolutions of raw videos, 1080p YUV sequences are resized into 720p YUV sequences. Note that the resizing operation does not introduce traces of lossy HEVC compression [23]. Then, raw videos are divided into two non-overlapping groups to generate samples for the training and testing phases, respectively. For each raw video, only the first 200 frames are considered and then split into two non-overlapping video clips (each video clip contains 100 frames). Finally, there are 32 and 20 raw video clips for generating training and testing samples, respectively.

Single compressed videos are obtained by encoding raw video clips with the bit rate (B_2) and GOP size (G_2). On the other hand, double compressed videos are obtained by first

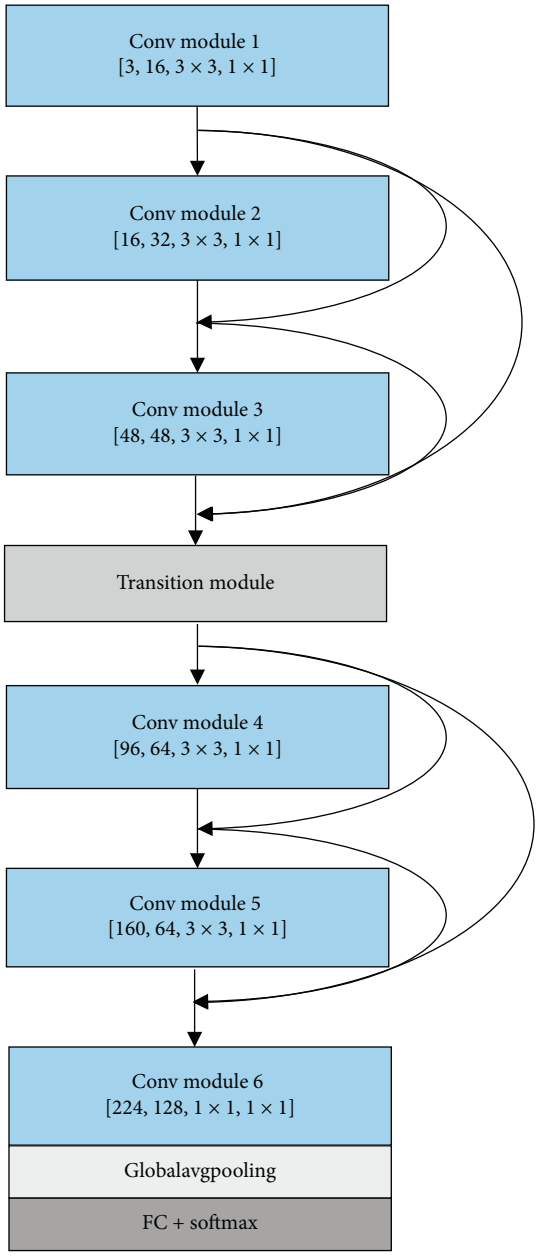

Figure 4: The overall network architecture of S-DenseNet. Detailed structures are illustrated in Section 4.3.

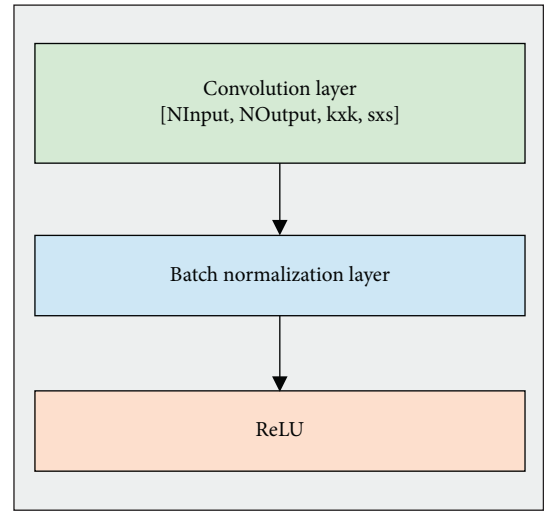

Figure 5: The detailed structure in a convolutional module. [NInput, NOutput, $k \times k$, $s \times s$] denotes the detailed settings of each convolutional layer, where NInput denotes the number of input feature maps; NOutput denotes the number of output feature maps; and $k \times k$ and $s \times s$ denote the $k \times k$ convolutional kernel with the stride $s \times s$.

encoding raw video clips with the bit rate (B_1) and GOP size (G_1). These single compressed videos are decompressed and then recompressed with the bit rate (B_2) and GOP size (G_2). B_1 and B_2 are both selected from $\{500, 1700, 3000\}$ kbps. We consider compression processes with the increased or decreased bit rates. G_1 and G_2 are selected from $\{14, 30\}$ and $\{9, 25, 70\}$, respectively. One of the most popular HEVC codecs, namely, ×265, is applied to conduct encoding and decoding processes with the main profile. Other coding parameters are set as default unless otherwise specified. Consequently, for the training phase, 288 single compressed HEVC videos and 1152 double compressed HEVC videos are obtained to construct the video set $\mathscr{V}_{\text{train}}$. For the testing phase, 180 single compressed HEVC videos and 720 double compressed HEVC videos are obtained to construct the video set $\mathscr{V}_{\text{test}}$.

As claimed in Section 4, the preprocessed CU information maps of a short-time video clip $\{\mathbf{C}_{t-1}', \mathbf{C}_t', \mathbf{C}_{t+1}'\}$ is treated as a sample, where N_b is set as 224. This resizing operation is conducted by Python Imaging Library with the default setting. Please note we only consider short-time video clips which do not contain re-encoded I-frames. We randomly select N_s samples whose middle frames are TypeI P-frames, N_s samples whose middle frames are TypeII P-frames, and $2 \times N_s$ samples whose middle frames are TypeIII P-frames from $\mathscr{V}_{\text{train}}$ to construct the set $\mathscr{T}_{\text{train}}$ of training samples, where N_s is set as 3500. In the same manner, the set $\mathscr{T}_{\text{test}}$ of testing samples can be constructed based on $\mathscr{V}_{\text{test}}$, where N_s is set as 4500. For simplicity, "TypeI P-frame (TypeII/TypeIII P-frame)" is used as the abbreviation of "a short-time video clip whose middle frame is TypeI P-frames (TypeII/TypeIII P-frame)" in following sections unless otherwise specified.

5.2. The Training and Testing Protocols

(i) To obtain the trained SVM classifier [28] in the first stage, all TypeI P-frames are treated as positive samples while N_1 samples ($N_1/2$ TypeII P-frames and $N_1/2$ TypeIII P-frames) are randomly selected from the set $\mathscr{T}_{\text{train}}$ as negative samples. N_1 is set as 3500. The optimal SVM classifier is obtained by applying five-fold cross validation.

(ii) To train the S-DenseNet in the second stage, all TypeII P-frames are treated as positive samples and N_2 TypeIII P-frames are randomly selected from the set $\mathscr{T}_{\text{train}}$ as negative samples. TypeIII P-frames include single compressed P-frames and P-P frames.

(a) (b)

(c)

Figure 6: Examples of raw videos. (a) Four people. (b) Old town. (c) Pedestrian.

N_2 is set as 3500. Then, N_2 pairs of samples are randomly divided into two parts (9:1) as the training set and validation set. In the training process, the weights in convolutional layers are initialized with the method in [29]. The learning rate is initialized as 0.0001 and reduced by 50% after every 6 training epochs. The mini-batch size is set as 16. The maximum number of epochs is set as 150, and the optimal S-DenseNet is obtained by achieving the best performance on the validation set. Experiments are conducted on a device equipped with CPU Intel Core i7-8700K and GPU GTX 1080 Ti.

(iii) In the testing phase, TypeI P-frames and TypeII P-frames (namely, relocated I-frames) in $\mathcal{T}_{\text{test}}$ are treated as positive samples while TypeIII P-frames in $\mathcal{T}_{\text{test}}$ are treated as negative samples. Detection accuracy is applied as the criterion to evaluate the detection performance, which can be formulated as follows:

$$\text{Acc} = \frac{1}{2}\left(\frac{\text{TP}}{P} + \frac{\text{TN}}{N}\right) \times 100\%, \tag{7}$$

where TP and TN denote the number of positive and negative samples classified correctly and P and N denote the total number of positive and negative samples.

5.3. Comparison Experiment. In this experiment, the proposed method is compared with several state-of-the-art methods, including a hand-crafted feature-based method [17] and a deep-learning-based method [21]. Samples in $\mathcal{T}_{\text{train}}$ and $\mathcal{T}_{\text{test}}$ are used to conduct training and testing phases, respectively. We briefly present some details of [17, 21] here.

(i) In [17], the authors extended the concept of VPF in [7] considering motion vectors. They constructed a feature named as generalized VPF (GVPF) to identify relocated I-frames. Following the same setting in [21], GVPF can be extended to HEVC videos as follows:

$$v_t^H = \begin{cases} 0, & \text{if } \forall b \in \mathcal{B}, g_{\mathbf{i},b}(t,1) = g_{-\mathbf{s},b}(t,1) = g_{\tilde{\mathbf{p}},b}(t,1) = 1, \\ \sum_{b\in\mathcal{B}} \sum_{k\in\mathcal{K}} g_{\mathbf{i},b}(t,k) g_{-\mathbf{s},b}(t,k) g_{\tilde{\mathbf{p}},b}(t,k), & \text{otherwise}, \end{cases} \tag{8}$$

$$g_{\mathbf{a}}(t,k) = \begin{cases} |a_t - a_{t-k}|, & \text{if } a_t > \max(a_{t-1}, a_{t+1}), \\ 1, & \text{otherwise}, \end{cases} \tag{9}$$

where $\mathscr{K} = \{-1, 1\}$ and $\mathscr{B} = \{8, 16, 32, 64\}$. In equation (9), a_t denotes the tth element in the vector $\mathbf{a}$. Specifically, $(\mathbf{i}, b)$,$(-\mathbf{s}, b)$, and $(\tilde{\mathbf{p}}, b)$ denote the vectors whose element is the number of different CU types with the block size $b \times b$ in a frame. For example, $(i, b)_t$ in $(\mathbf{i}, b)$ denotes the number of I-CU with the block size $b \times b$ in tth frame. More details can be found in [17, 21]. Then, the extended GVPF (v_t^H) is passed into the threshold-based classifier to detect relocated I-frames. The threshold is determined by obtaining the best performance on the validation set.

(ii) In [21], the authors proposed a hybrid network to learn spatiotemporal representations of relocated I-frames in compression domain. In this hybrid network, an attention-based two-stream ResNet is designed to extract spatial patterns and LSTM is used to capture temporal variation of coding information in compression domain. The optimal model is obtained by achieving the best detection result on the validation set. Other experimental settings are identical to [21].

5.3.1. Detection Accuracies with Different Bit Rates. Bit rate is a primary factor to control the video quality in the encoding process. As mentioned in Section 3.2, the statistics of CU's information have quite different behaviors with various recompression bit rates. It is valuable to evaluate detection performance of the proposed method and other state-of-the-art methods with different bit rate combinations (B_1, B_2). Experimental results are presented in Table 1.

As shown in Table 1, the proposed method achieves the best detection result in all cases compared with other state-of-the-art methods. The promising detection performance of the proposed method verifies the superiority of applying the two-stage cascaded scheme to deal with various recompression scenarios. The performance improvement is distinct when the difference of bit rates in single and double compression (namely, $\Delta B = |B_1 - B_2|$) is larger than 2000 kbps. This result infers that it is proper to apply individual detectors to expose relocated I-frames with different recompression bit rates instead of a universal detector, which is also in accordance with the analysis of CU statistics in Section 3.2. On the other hand, the poorer detection result of GVPF [17] when $B_1 > B_2$ indicates that the low-order statistics of CU types are insufficient to discriminate the slight traces between TypeII P-frames and TypeIII P-frames. Besides, for all methods, the detection performance becomes better with the increment of recompression bit rate (B_2), since the degradation of lossy quantization is slighter in this case.

The time efficiency of the proposed method is evaluated in this experiment. 1000 samples are randomly selected from the testing set. More specifically, a set of CU information maps from a short-time video clip denotes a sample. Consequently, it takes 92 ms and 35 ms to process one sample on average for the first and second stages, respectively.

5.3.2. Detection Accuracies with Different GOP Sizes. In a GOP unit, the initial frame is the intra-coded frame and the rest of the frames are inter-coded frames. Due to the existence of error propagation during the motion compensation, GOP size is another significant factor which has great influence on the visual quality of compressed videos. In this experiment, the detection performance is evaluated with different GOP combinations (G_1, G_2). Experimental results are presented in Figure 7.

As shown in Figure 7, the proposed method can achieve the distinct improvement compared with other state-of-the-art methods, especially when the GOP size is very small in the recompression process (e.g., the proposed method improves more than 7% of detection accuracies for $G_2 = 9$). With the smaller G_2, the detection of relocated I-frames becomes more challenging, since the inconsistency between P-frames in different GOP units during single compression is easier to be degraded by the more frequent occurrence of intra-coded frames in recompression. The promising result of the proposed method verifies the advantage of applying individual detectors to handle different recompression scenarios. Besides, all methods can perform better with the increment of G_2 due to the less influence of the intra-coding process during recompression.

5.4. Analyzing Different Network Architectures. To discriminate TypeII and TypeIII P-frames, we design a shallow CNN equipped with dense connections which can jointly learn spatiotemporal representations from low-level patterns in early layers and high-level forensics semantics in top layers. It is valuable to study how different network architectures influence the detection performance. In this experiment, we consider the following network architectures. (1) Plain CNN: all dense connections are removed from S-DenseNet. The number of output feature maps in each convolutional layer keeps unchanged and the number of input feature maps in its next module is modified correspondingly. (2) S-DenseNet-5: the fourth conv module is removed and dense connections are modified correspondingly. Other network architectures keep unchanged. (3) S-DenseNet-7: A conv module is added after the original fourth conv module. The dense connections are also added between this new conv module and other conv modules. The detection accuracy is calculated using all testing samples which are obtained from recompressed videos with the decrease bit rate in $\mathscr{T}_{\text{test}}$. Other experimental settings are identical to those in Section 5.3.1.

It can be observed in Table 2 that the average detection accuracy suffers from a distinct decrease (more than 2%) when the dense connections are removed. Based on the results of networks equipped with dense connections, adding dense connections can help the network learn spatiotemporal representations more efficiently via feature reuse in early layers. On the other hand, the relatively worse detection results of S-DenseNet-5 and S-DenseNet-7 demonstrate that reducing the number of conv modules may lead to insufficient learning capability while adding extra conv modules has the risk of overfitting. In summary, dense connections are helpful to achieve a promising performance gain and it is also very important to construct the S-DenseNet with a proper number of conv modules for frame-wise detection of double HEVC compression.

Table 1: The detection accuracies with different bit rate combinations (%).

	Recompression with the increased bit rate			Recompression with the decreased bit rate		
(B_1, B_2)kbps	(500, 3000)	(500, 1700)	(1700, 3000)	(3000, 1700)	(1700, 500)	(3000, 500)
GVPF [17]	96.52	95.43	94.84	86.90	77.35	67.86
CNN-LSTM [21]	95.35	95.19	96.23	91.38	83.57	74.51
Proposed	**100**	**99.07**	**97.38**	**92.59**	**85.99**	**83.50**

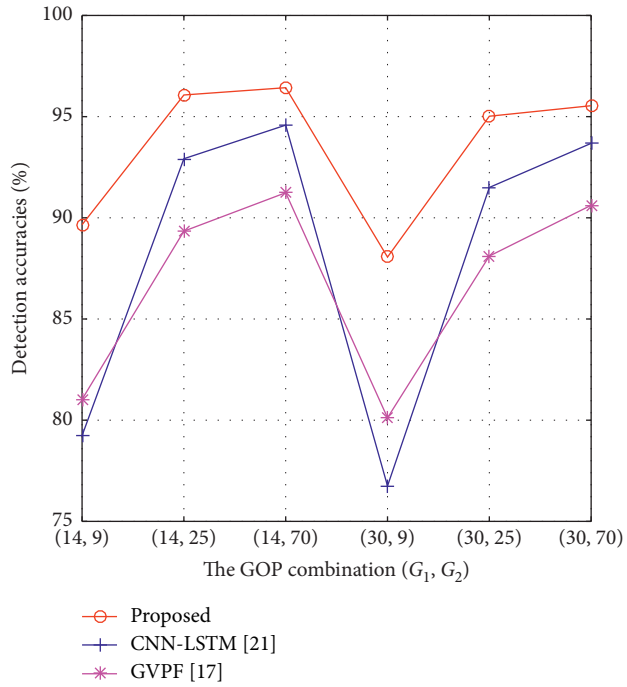

Figure 7: The detection accuracies with different GOP combinations (G_1, G_2).

Table 2: Detection performance with different network architectures (%).

Network architectures	Plain CNN	S-DenseNet-5	S-DenseNet-7	Proposed
Accuracy	86.42	87.15	88.23	**88.51**

We also evaluate the detection performance of the S-DenseNet by replacing the average pooling operation in the transition module by the max-pooling operation. There is a slight performance drop (about 0.75% on average) of the modified network, which implies that the average pooling operation is more suitable to process CU information maps containing potential variations of CU types in local regions. Besides, based on experimental results, the proposed method has a slight performance drop (about 0.55%) when the resizing operation is not applied for CU information maps in the second stage. It infers that the proposed S-DenseNet may learn deep representations more efficiently from input samples whose elements have more diverse statuses.

5.5. Performance Evaluation with Different Rate-Distortion Strategies. For each encoder, a proper rate-distortion strategy (referred to as RDO strategy) is applied to balance the quality of compressed videos and the computational complexity. In practical applications, the suspicious video may be encoded with a RDO strategy which is not considered in the training phase of detectors. Different RDO strategies can cause dissimilar properties of coding information in compression domain. Therefore, it is significant to analyze how different methods perform with the mismatched RDO strategy in the testing phase. In this experiment, single and double compressed videos are generated with the option "–rd" set as 5 to construct the video set $\mathscr{V}_{\text{test\_RDO}}$. Then, testing samples are generated using videos in $\mathscr{V}_{\text{test\_RDO}}$ in the same way as Section 5.2. In ×265, the higher value of "–rd" means that a more complicated and exhaustive RDO strategy is used to achieve better visual quality with a fixed bit rate. In Section 5.3.1, the option "–rd 3" is used as the default setting to generate training samples. The trained models in Section 5.3.1 are used to obtain detection results directly in this experiment. The detection results are presented in Table 3.

As shown in Table 3, the proposed method still achieves the promising detection result when single compressed videos are recompressed with the increased or decreased bit rates. Compared with the results in Table 1, the detection accuracies of the proposed method suffer from a slight decrease (less than 2%) in a few cases due to the different statistics of coding information caused by the unknown RDO strategy of $\mathscr{V}_{\text{test\_RDO}}$. The more robust detection capability verifies our detection feature combined with the SVM classifier in the first stage, and a shallow CNN equipped with dense connections in the second stage is efficient to capture unique temporal inconsistency for different recompression scenarios. On the other hand, testing samples generated by the unknown RDO strategy are easier to have negative impact on the detection performance of the deep-learning-based method which only constructed a universal detector [21].

5.6. Performance Evaluation with Transcoding Process. When videos are transmitted over the communication networks, they are likely to be recompressed with a video coding standard different from the original one. This kind of recompression process is called as heterogeneous transcoding process. It is significant for double compression detection methods to be robust against the transcoding process. In this experiment, we consider a common transocding process in current network communications, namely, H.264 videos re-encoded as HEVC videos. To generate testing samples, raw video clips are first compressed using lib × 264 which is a widely used H.264/AVC codec. Then, these single compressed videos are recompressed with ×265. Other coding parameters, including bit rates and

Table 3: The detection accuracies with different rate-distortion strategies (%).

	Recompression with the increased bit rate			Recompression with the decreased bit rate		
(B_1, B_2)kbps	(500, 3000)	(500, 1700)	(1700, 3000)	(3000, 1700)	(1700, 500)	(3000, 500)
GVPF [17]	93.78	92.35	92.11	85.67	76.96	65.95
CNN-LSTM [21]	92.66	93.94	95.79	89.86	82.48	74.17
Proposed	**98.15**	**97.45**	**96.21**	**91.12**	**86.82**	**81.55**

Table 4: The detection accuracies with the transcoding process (%).

	Recompression with the increased bit rate			Recompression with the decreased bit rate		
(B_1, B_2)kbps	(500, 3000)	(500, 1700)	(1700, 3000)	(3000, 1700)	(1700, 500)	(3000, 500)
GVPF [17]	97.46	95.53	96.83	85.97	78.69	70.46
CNN-LSTM [21]	96.92	97.59	96.58	93.15	84.74	81.09
Proposed	**99.88**	**99.53**	**97.45**	**96.64**	**89.43**	**87.21**

GOP sizes, are equal to the settings in Section 5.3.1. Trained detection models in Section 5.3.1 are applied to obtain detection results directly without retraining. The detection results of different methods are presented in Table 4.

It can be observed that the proposed method still achieves the best detection result compared with other state-of-the-art methods when the heterogeneous transcoding process is conducted. Both the proposed method and CNN-LSTM [21] achieve some performance gain of detection accuracies compared with the results in Table 1, especially when $B_1 > B_2$. The reason behind this is that the block partition strategies applied in the single compression (the fixed 16×16 macroblock in H.264/AVC) and the double compression (the flexible partition strategy in HEVC) can cause the mismatch of block boundaries, which makes temporal inconsistency of recompression more distinct.

5.7. Performance Evaluation with Unknown Coding Parameters. In practical applications, suspicious videos may be encoded with unknown coding parameters which are unseen in the training phase. In this experiment, detection capability of different methods against unknown coding parameters, including bit rate and GOP size, is evaluated. The generation process of testing samples is the same as that in Section 5.1, except that bit rate and GOP size are set as different values. More specifically, B_1 and B_2 are selected from the set $\{800, 1500, 2500\}$kbps; G_1 is selected from $\{18, 34\}$; and G_2 is selected from $\{12, 45\}$. Other experimental settings are identical to those in Section 5.1. The trained models in Section 5.3.1 are applied to obtain the detection results directly in this experiment without retraining. The detection results are presented in Table 5.

As shown in Table 5, the proposed method can achieve reliable detection results for testing samples generated by unknown coding parameters, including bit rate and GOP size. Besides, the detection accuracies of the proposed method and CNN-LSTM [21] obtain a slight improvement compared with the results in Section 5.3.1. It may be due to fact that the difference between the bit rates applied in single and double compression (namely, $\Delta B = |B_1 - B_2|$) is smaller in this experiment than the coding parameter setting in the training phase, which leads to more stable and detectable traces of double compression. The robust detection capability against unknown coding parameter settings is very important in practical forensic applications.

5.8. Performance Evaluation with Different Video Resolutions. With the development of video compression technologies and video capture devices, people are more easily accessing digital videos with high resolutions (such as 1080p and 4K) in practical applications. It is significant to evaluate the detection capability of the proposed method for different resolutions. In this experiment, raw video sequences (namely, YUV sequences) with different resolutions, including 1080p and 4K, are collected from the Internet to generate single and double compressed videos. The downloading address and the list of raw video sequences are presented in the Appendix. According to the same manner mentioned in Section 5.1, for 1080p YUV sequences, bit rates are selected from {1000, 3400, 6000}kbps and other coding parameters keep unchanged to generate the set of testing samples denoted as $\mathcal{T}_{\text{test\_1080p}}$. On the other hand, for 4K YUV sequences, bit rates are selected from {5000, 17000, 30000}kbps and other coding parameters keep unchanged to generate the set of testing samples denoted as $\mathcal{T}_{\text{test\_4K}}$. The increment of bit rates is due to the higher video resolutions. To conduct the detection using the trained models in Section 5.3 directly, the 180×320 central regions of CU information maps in each sample are cropped to align the resolution of samples used to train the original models. Following the testing protocol mentioned in Section 5.2, the average detection accuracies of different methods are presented in Table 6.

As shown in Table 6, it can be observed that the proposed method can achieve outstanding detection results for various resolutions, such as 1080p and 4K, among different detection methods. There are some drops of detection accuracies for all methods compared with their results for 720p videos. This phenomenon is reasonable due to the following reasons: (1) we do not train new detection models from scratch using samples with different resolutions and (2) only part of coding information is used to align the resolution of samples

Table 5: The detection accuracies with unknown coding parameters (%).

	Recompression with the increased bit rate			Recompression with the decreased bit rate		
(B_1, B_2)kbps	(800, 2500)	(800, 1500)	(1500, 2500)	(2500, 1500)	(1500, 800)	(2500, 800)
GVPF [17]	96.23	95.13	95.66	87.34	81.75	75.62
CNN-LSTM [21]	95.19	94.29	96.37	91.21	87.73	85.03
Proposed	**99.33**	**96.87**	**97.10**	**93.58**	**92.97**	**90.07**

Table 6: The detection accuracies with different resolutions (%).

	Recompression with the increased bitrate			Recompression with the decreased bitrate		
Video resolution	720p	1080p	4K	720p	1080p	4K
GVPF [17]	95.36	93.42	93.67	76.57	72.03	71.29
CNN-LSTM [21]	95.59	92.75	92.88	84.15	80.33	78.48
Proposed	**98.81**	**96.17**	**96.32**	**88.51**	**87.92**	**86.40**

used to train the original models. The more reliable detection capability encourages us that it is possible to detect frame-wise double HEVC compression with higher resolutions (such as 8K) in future work, by constructing the detection model with low-resolution input samples, and then obtain final detection results using a proper fusion strategy.

6. Conclusions

In this work, by analyzing the statistics of CU information (block size and prediction mode), we found that the distributions of different CU types' ratios have dissimilar properties when original videos are recompressed by the increased and decreased bit rates. It motivates us to design a two-stage cascaded detection scheme for double HEVC compression based on temporal inconsistency, which aims to provide more reliable detection capability for various recompression bit rates in practical applications. For a given video, the CU information map of a short-time video clip is first constructed. In the first stage of detection, a compact detection feature is extracted based on the distributions of different CU types and the K-L divergence of adjacent frames. Then, this detection feature is fed into the trained SVM classifier to identify TypeI P-frames. In the second stage, a shallow CNN equipped with dense connections is carefully designed to extract spatiotemporal representations from coding information in compression domain to obtain final detection results. In experiments, the proposed detection scheme achieves a distinct improvement, especially when the difference between the original bit rate and the recompression bit rate is dramatic (e.g., $\Delta B = |B_1 - B_2| > 2000$kbps). Besides, the detection performance of the proposed method is reliable under various scenarios, such as mismatched rate-distortion strategy, unknown coding parameters, and transcoding process. This advantage is significant in real forensic applications. In future study, we will extend this work in the following aspects: (1) consider the unique coding units in B-frames, such as bi-prediction CU, and (2) leverage other kinds of coding information in CU to construct CU information map, such as merge mode.

Appendix

Same as the setting in [23], the list of YUV sequences: 720p: ducks-take-off, old-town-cross, park-joy, FourPeople, in-to-tree, Johnny, KristenAndSara, mobcal, vidyo1, vidyo3, parkrun, Stockholm, and shields; 1080p: blue-sky, riverbed, crowd-run, pedestrian-area, rush-field-cuts, rush-hour, speed-bag, snow-mnt, touchdown-pass, tractor, west-wind-easy, station2, and sunflower. The downloading address is https://media.xiph.org/video/derf/.

The list of YUV sequences for training samples: 720p: ducks-take-off, Johnny, KristenAndSara, shields, Stockholm, and vidyo3; 1080p: blue-sky, crowd-run, pedestrian-area, rush-hour, rush-field-cuts, speed-bag, station2, sunflower, touch down-pass, and west-wind-easy. The list of YUV sequences for testing samples: 720p: FourPeople, in-to-tree, mobcal, old-town-cross, park-joy, parkrun, and vidyo1; 1080p: riverbed, snow-mnt, and tractor.

In Section 5.8, the downloading address is https://media.xiph.org/video/derf/. The list of YUV sequences used to generate testing samples: (1) 1080p: aspen, controlled-burn, dinner, ducks-take-off, in-to-tree, old-town-cross, park-joy, riverbed, snow-mnt, and tractor; (2) 4K: Netflix-FoodMarket2, Netflix-SquareAndTimelapse, Netflix-Boat, Netflix-Food Market, Netflix-Tango, Netflix-BoxingPractice, Netflix-Narrator, Netflix-TunnelFlag, Netflix-Crosswalk, and Netflix-RitualDance.

Data Availability

The download address of raw videos is https://media.xiph.org/video/derf/.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This study was supported by the National Natural Science Foundation of China (61902263 and 61972269), China Postdoctoral Science Foundation (2020M673276), and Fundamental Research Funds for the Central Universities (2020SCU12066 and YJ201881).

References

[1] C.-X. Wang, F. Haider, X. Gao et al., "Cellular architecture and key technologies for 5G wireless communication networks," *IEEE Communications Magazine*, vol. 52, no. 2, pp. 122–130, 2014.

[2] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[3] W. Chen and Y. Q. Shi, "Detection of double MPEG compression based on first digit statistics," in *Digital Watermarking*, pp. 16–30, Springer, Berlin, Germany, 2009.

[4] X. Jiang, W. Wang, T. Sun, Y. Q. Shi, and S. Wang, "Detection of double compression in MPEG-4 videos based on Markov statistics," *IEEE Signal Processing Letters*, vol. 20, no. 5, pp. 447–450, 2013.

[5] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double MPEG compression," in *Proceedings of the 8th Workshop On Multimedia and Security*, Geneva, Switzerland, April 2006.

[6] M. C. Stamm, W. S. Lin, and K. J. R. Liu, "Temporal forensics and anti-forensics for motion compensated video," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 4, pp. 1315–1329, 2012.

[7] D. Vazquez-Padin, M. Fontani, T. Bianchi, P. Comesaña, A. Piva, and M. Barni, "Detection of video double encoding with GOP size estimation," in *Proceedings of the Information Forensics and Security (WIFS), 2012 IEEE International Workshop on*, pp. 151–156, Tenerife, Spain, December 2012.

[8] W. Luo, M. Wu, and J. Huang, "MPEG recompression detection based on block artifacts," in *Electronic Imaging*International Society for Optics and Photonics, Bellingham, Washington USA, 2008.

[9] P. He, T. Sun, X. Jiang, and S. Wang, "Double compression detection in MPEG-4 videos based on block artifact measurement with variation of prediction footprint," in *Proceedings of the International Conference on Intelligent Computing*, pp. 787–793, Fuzhou, China., August 2015.

[10] P. He, X. Jiang, T. Sun, and S. Wang, "Detection of double compression in MPEG-4 videos based on block artifact measurement," *Neurocomputing*, vol. 228, pp. 84–96, 2017.

[11] P. He, X. Jiang, T. Sun, S. Wang, B. Li, and Y. Dong, "Frame-wise detection of relocated I-frames in double compressed H.264 videos based on convolutional neural network," *Journal of Visual Communication and Image Representation*, vol. 48, pp. 149–158, 2017.

[12] S. Nam, J. Park, D. Kim, I. Yu, T. Kim, and H. Lee, "Two-stream network for detecting double compression of H.264 videos," in *Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP)*, pp. 111–115, Taipei, Taiwan, September 2019.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 26th Annual Conference on Neural Information Processing Systems*, pp. 1106–1114, Sydney, Australia, December 2012.

[14] X. Zhang, T. Huang, Y. Tian, M. Geng, S. Ma, and W. Gao, "Fast and efficient transcoding based on low-complexity background modeling and adaptive block classification," *IEEE Transactions on Multimedia*, vol. 15, no. 8, pp. 1769–1785, 2013.

[15] S. Bian, W. Luo, and J. Huang, "Exposing fake bit rate videos and estimating original bit rates," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 12, pp. 2144–2154, 2014.

[16] S. Milani, M. Fontani, P. Bestagini et al., "An overview on video forensics," *APSIPA Transactions on Signal and Information Processing*, vol. 1, no. 2, 2012.

[17] D. Vázquez-Padín, M. Fontani, D. Shullani, F. Pérez-González, A. Piva, and M. Barni, "Video integrity verification and GOP size estimation via generalized variation of prediction footprint," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 9, 2019.

[18] X. Jiang, P. He, T. Sun, and R. Wang, "Detection of double compressed HEVC videos using GOP-based PU type statistics," *IEEE Access*, vol. 7, pp. 95364–95375, 2019.

[19] S. Bian, W. Luo, and J. Huang, "Exposing video forgeries by detecting misaligned double compression," in *Proceedings of the International Conference on Video and Image Processing*, pp. 44–48, Singapore, December 2017.

[20] Q. Wu, T. Sun, X. Jiang, K. Xu, Q. Xu, and P. He, "HEVC double compression detection with non-aligned GOP structures based on a fusion feature with optical flow and prediction units," in *Proceedings of the 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI)*, pp. 1–6, Jiangsu, China, October 2019.

[21] P. He, H. Li, H. Wang, S. Wang, X. Jiang, and R. Zhang, "Frame-wise detection of double HEVC compression by learning deep spatio-temporal representations in compression domain," *IEEE Transactions on Multimedia*, vol. 21, 2020.

[22] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

[23] P. He, H. Li, B. Li, H. Wang, and L. Liu, "Exposing fake bitrate videos using hybrid deep-learning network from recompression error," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 4034–4049, 2020.

[24] Y. Yu, H. Yao, R. Ni, and Y. Zhao, "Detection of fake high definition for HEVC videos based on prediction mode feature," *Signal Process*, vol. 166, 2020.

[25] X. Jiang, Q. Xu, T. Sun, B. Li, and P. He, "Detection of HEVC double compression with the same coding parameters based on analysis of intra coding quality degradation process," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 250–263, 2019.

[26] N. C. Schwertman, M. A. Owens, and R. Adnan, "A simple more general boxplot method for identifying outliers," *Computational Statistics & Data Analysis*, vol. 47, no. 1, pp. 165–174, 2004.

[27] G. Huang, Z. Liu, L. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HW, USA, July 2017.

[28] C. Chang and C. L. Libsvm, "A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1026–1034, Los Condes, CL, USA, December 2015.