



Double compression detection in HEVC-coded video with the same coding parameters using picture partitioning information

Kutub Uddin, Yoonmo Yang, Byung Tae Oh *

Korea Aerospace University, Republic of Korea

ARTICLE INFO

Keywords:

Video forensic
Deep learning
HEVC
Picture partitioning
Double compression detection

ABSTRACT

Detection of double compression, particularly in the high-efficiency video coding (HEVC) compressed domain, is one of the most operative and efficacious ways of authenticating the validity of videos in the field of forensic analysis. The strength of identifying abnormalities in the videos confides in diverse coding parameters (such as quantization parameters, size and structure of the group of pictures, and modes of compression). Many methods have been introduced to dig up HEVC double compression with different coding parameters. However, the revelation of the HEVC double compression under the same coding environments still remains a competitive task, as recompressions leave small footprints. In this paper, we introduce a novel method based on frame partitioning information to distinguish between single and double compressions with the same coding parameters. We propose extracting statistical and deep convolution neural network (DCNN) features from partition pictures and prediction modes, including coding unit, prediction unit, transform unit, and most probable modes information. Finally, machine learning technology is integrated to categorize videos into two classes, single and double compressions, by combining the statistical and DCNN features. We obtain the best experimental results by assembling the statistical and DCNN features for wide video graphics array (WVGA) and high-definition (HD) sequences with average accuracies of 99.66% and 99.60% in all-intra and 99.46% and 99.33% in low-delay P modes respectively. Experimental results of the proposed system show the effectiveness and efficiency over the state-of-the-art techniques in video forensic.

1. Introduction

Multimedia information such as image, audio, and video are becoming essential in everyday lives, as smart surveillance systems and digital cameras have improved with larger storage capacity and better quality. Digital video technology is a medium of communication, helping us share important moments on social networks such as Facebook, Twitter, and YouTube. These videos reflect our day-to-day activities. Sometimes, these videos can harm our dignity and honor, as video forgers are becoming experts in the latest video technology. Moreover, video editing software applications can rapidly manipulate videos without leaving substantial evidence. With the engagement of expert counterfeiters in video forgery, it is difficult to authenticate the integrity of videos. Even an amateur can easily forge videos without leaving any visual trace using advancements in video technology such as partial frame deletion introduced by Hong et al. [1]. Milani et al. [2] described video forensic analysis of the acquisition, copy-paste, compression, and other editing operations.

With the increase in high-quality video capturing devices, it is becoming more difficult to store and transmit videos. Video codecs were

improved to reduce the spatial and temporal redundancy in videos, particularly in high-quality videos. High-efficiency video coding (HEVC) is the most recent standard video codec for the compression of high-definition (HD) videos to ensure less storage and faster transmission over the devices. HEVC is a lossy compression technique that reduces the spatial and temporal redundancy in videos by introducing several new approaches (particularly quad-tree-based picture partitioning and flexible intra prediction modes). During the manipulation of HD videos, HEVC double compression is unavoidable [3,4]. Thus, HEVC double compression may be an indication of forgery in video content. Detection of double compression can help establish the authenticity of the videos and identify video manipulation for malicious purposes.

Several methods have been proposed to detect double compression in images and videos. As an early approach, double compression detection in Joint Photographic Experts Group (JPEG) images has been widely researched. The detection of double JPEG compression with the same quantization matrix is a distinct approach. Huang et al. [5] analyzed and suggested a method for detecting double JPEG compression with the same environments based on a random perturbation strategy. Yang et al. [6] reported two types of errors (rounding and truncation errors) to detect the single and double JPEG compressions with the same

* Correspondence to: 76 Hanggongdaehang-ro, Deogyang-gu, Goyang, Republic of Korea.

E-mail address: byungoh@kau.ac.kr (B.T. Oh).

quantization matrix. They exposed the errors occurring in multiple compressions of the JPEG images between the inverse discrete cosine transform and rounding-truncation images. Huang et al. [7] proposed a method for detecting double JPEG compression by extracting two types of features, error-based statistical, and convolutional neural network (CNN) features, based on the rounding and truncation errors in images. Barni et al. [8] proposed to detect double JPEG compression using CNN in aligned and non-aligned cases. They explored the capability of CNN to identify double JPEG compression directly from images. Another CNN-based method was proposed based on rounding-truncation errors in images extracted from the input sample, and the performance was evaluated with a set of quality factors [9] by Peng et al. in the JPEG compressed domain with the same quantization matrix.

Similar approaches to detect double JPEG compression were used in the Moving Picture Expert Group (MPEG)-2 intra coding process to detect the double compression. Sun et al. [10] extracted 12-D features from the first digit discrete cosine transform (DCT) coefficient for each group of pictures (GOP) based on the violation of the parametric logarithm law for detecting the double compression in the MPEG format. Aghamaleki et al. [11] used a quantization matrix and synchronized GOP for efficient detection of double compression in MPEG videos. Jiang et al. [12] extracted Markov-based features to detect the artifacts of double compression in MPEG-4. Some methods focused on detecting H.264/advanced video coding (AVC) double compression to reveal the manipulations. Liu et al. [13] introduced detection of frame deletion in H.264/AVC video codec using the average residual of P-frames, its time, and frequency domain. Yao et al. [14] proposed double compression detection in H.264/AVC as the preliminary idea for detecting video manipulation by using the periodic feature of the data bit string and the skip macroblocks. Double compression detection in H.264/AVC was suggested by Liao et al. [15] using the probability distribution of quantized non-zero AC coefficients.

With the evolution of HEVC, the detection of double compression in the HEVC compressed domain was also inspired by a number of researchers. Xu et al. [16] used the sequence of number of prediction unit of its prediction mode (SN-PUPM) feature from each frame of a video in three different modes (“Intra”, “Inter”, and “Skip”) to disclose tampering by detecting HEVC double compression. We introduced several algorithms [16–21] to detect HEVC double compression with different parameter settings for uncovering the manipulation of the videos to ensure integrity and authenticity. Li et al. [17] analyzed two subsets of features from the probabilities of quantized DCT (QDCT) coefficients and transform unit (TU) size. First, a 9-D feature was calculated from the QDCT coefficients, and the remaining 8-D feature was calculated from 32×32 , 16×16 , 8×8 , and 4×4 TU size. The histogram feature extracted for the TU depth from I/P frames of a video to distinguish between single and double compressed videos under different bit rates was proposed by Yu et al. [18]. Huang et al. [19] and Li et al. [20] selected a 4×4 co-occurrence matrix of prediction unit (PU) and DCT coefficients to separate single and double compressed videos in the HEVC compressed domain. Jiang et al. [21] focused on the abnormality of the PU to expose the HEVC double compression.

In this study, we aim to detect the artifacts left during multiple compressions in HEVC compressed domain with the same coding configurations. We first analyze the effects of the coding unit (CU), PU, TU, and most probable modes (MPM) information on the video contents while performing multiple compressions. We propose, two subsets of feature vectors, including the statistical and deep convolution neural network (DCNN) features based on the CU, PU, TU, and MPM information by multiple compressions. Finally, we use feature vectors in separately and in combination to determine whether the video is single or double compressed in a frame-wise manner. We also investigate the sequence-wise detection-results to demonstrate the effectiveness of the proposed method.

The remainder of this paper is organized as follows. An overview of the HEVC standard is presented in Section 2. In Section 3, we

describe the related systems proposed over the years and explain double compression detection in HEVC with the same coding environments. A detailed description of the proposed method, including the statistical and DCNN features extraction, is provided in Section 4. Section 5 presents the experimental results by combining performance evaluation and analysis. In Section 6, we present the conclusion of the proposed research.

2. Overview of HEVC

Digital video capturing devices have improved in terms of storage and quality over the past years. With the introduction of high-quality videos, such as HD and Ultra HD, it is more challenging to encode these videos at the desired bit rate. The HEVC video compression standard, a successor of H.264/AVC, has been introduced with many new capabilities, including quad-tree-based picture partitioning and flexible intra prediction modes. HEVC was finalized by the ITU-T Video Coding Experts Group jointly with ISO/IEC MPEG in January 2013.

2.1. Partitioning of coding tree unit based on the quad-tree structure

HEVC introduces four basic units: coding tree unit (CTU), CU, PU, and TU, during a flexible and efficient picture partitioning. It encodes a video in a frame-by-frame manner, and each frame is divided into multiple equal-sized non-overlapping CTUs [22,23]. Each CTU contains luminance, chrominance, and syntax elements. A luminance coding tree block has a rectangular shape of $L \times L$ samples, and the chrominance has a size $L/2 \times L/2$. An encoded syntax element is responsible for specifying the value of L that can be equal to 16, 32, or 64. It is also possible to have a larger CTU that is more efficient when compressing high-resolution sequences. Each CTU is further split into CUs by building a quad-tree structure [22]. Fig. 1(a) and (b) depict an example of a general CTU partitioning scenario and a quad-tree structure for processing the CUs. A CTU of size 64×64 is recursively divided into four equal-sized CUs of size 32×32 , and a CU of size 32×32 is further divided into 16×16 , 8×8 , or 4×4 PUs based on the prediction mode. If a CTU has a size $2N \times 2N$, there are two types of intra-CU partition mode, $2N \times 2N$ or $N \times N$. This means a CTU can be either a single CU or split into four smaller equal-sized CUs. In Fig. 1(b), the white circle represents that the CTU or CU is split into four equal-sized CUs, and the black circle represents that the CU is not further split. The digits from 1 to 13 in Fig. 1(a) indicate the numbers of split CUs and the processing order of those CUs.

2.2. Intra-frame prediction in HEVC

Three different CU prediction modes exist in HEVC: (1) skip mode, (2) intra mode, and (3) inter mode. In the skip mode, the motion vector difference and residual energy are equal to zero. The CU prediction mode determines whether it uses intra-frame (spatial) or inter-frame (temporal) prediction. Each CU is further split into one, two, or four PUs for its prediction. HEVC uses prediction modes to remove the spatial redundancy. Intra-frame prediction is carried out based on the neighboring pixels to predict a new block. There are thirty-five different intra-prediction modes in HEVC to predict a prediction block (PB). Modes 0 and 1 indicate the planar and DC modes respectively. The mean of two linear predictions from four corner reference samples is used to predict the intra planar mode. The intra-DC mode uses the average of the reference samples. The rest of the 2 to 32 modes are intra-angular modes that perform the prediction based on 33 directions. Fig. 2(a) shows the details of the thirty-three intra prediction modes with corresponding angles used in HEVC.

A residual block is predicted based on the PU partition types and split into multiple TUs for transformation and quantization by building a quad-tree structure called a residual tree or transform tree [22], as shown in Fig. 2(b).

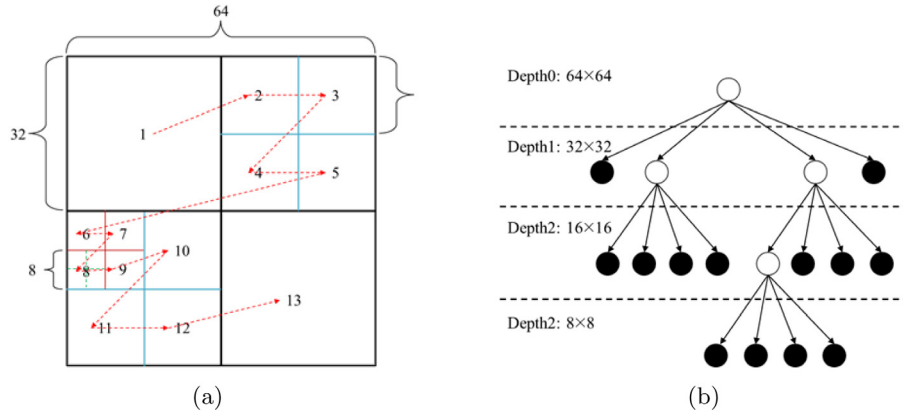


Fig. 1. Example of 64×64 CTU partition into CUs; (a) CTU partitioning; (b) quad-tree structure.

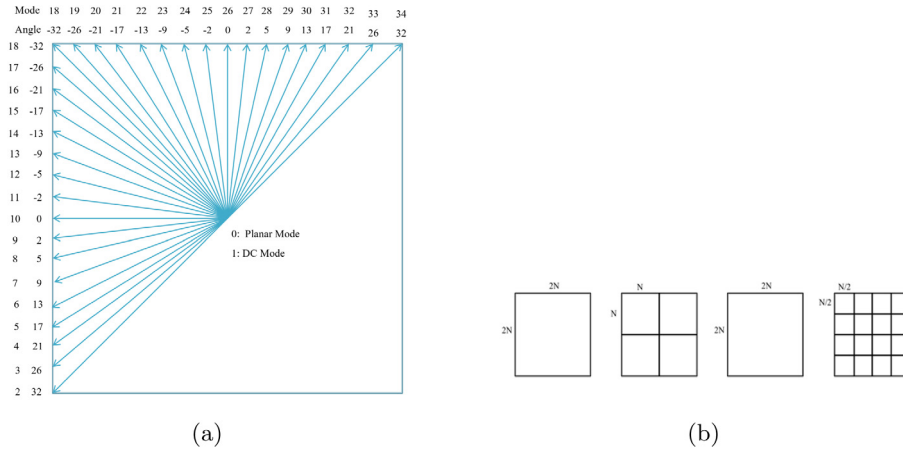


Fig. 2. Example of (a) intra-picture prediction and (b) TU partitioning in HEVC.

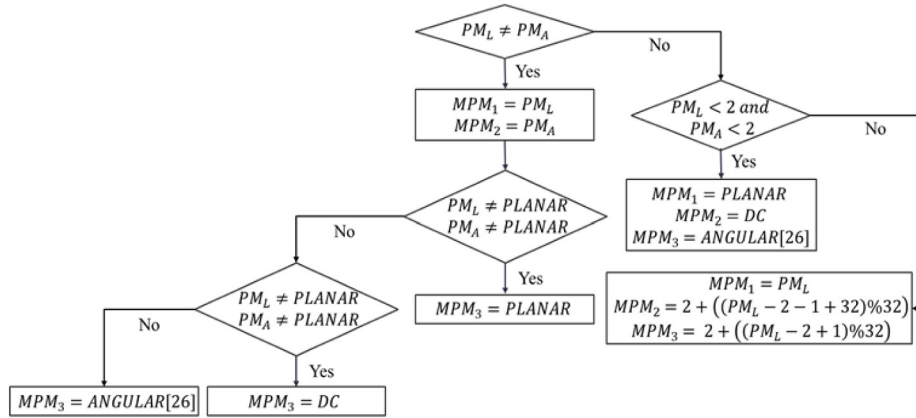


Fig. 3. Flow-chart of selection of three MPMs.

2.3. MPMs selection process

As the number of intra prediction modes increased in HEVC compared with H.264/AVC to improve prediction, the overhead also went up. An efficient intra prediction mode selection is important to reduce the overhead in intra-frame coding. Instead of using one MPM in H.264/AVC, HEVC derives the three MPMs for each prediction unit. The MPMs for the current PU are chosen based on the left PU's prediction mode (PM_L) and the above PU's prediction mode (PM_A). The flowchart for determining the three MPMs is shown in Fig. 3.

If the prediction mode of the current PU is chosen from any one of the MPMs, then, only the position is sent to the decoder. Otherwise, a 5-bit codeword is passed to indicate the rest of the prediction modes. We mainly focus on the CU, PU, TU, and MPM information for the proposed detection system to determine whether the video is single or double compressed.

3. Related work

With the development of more efficient software for performing video forgery, several efficient algorithms have also been developed to

detect double compression in the HEVC compressed domain to enhance the integrity of the video content under the different coding environments. We can observe more artifacts in the videos while conducting multiple compressions in HEVC-coded videos with different parameter settings than with the same parameter settings. Thus, it is easy to identify the unnaturalness in the recompressed videos with different coding parameter settings using footprints. However, this turns out to be a challenging task to detect double compression whenever we commit recompressions in the same coding environments. This is because there exist very small footprints in the recompressed videos under the same configuration settings. In this section, we overview related research for exposing the video forgery by detecting double compression.

Detection of double compression in the intra coding mode of the videos with the same coding configurations is similar to that of JPEG double compression detection with the same quantization matrix. Studies show that quality decreases between n th and $(n+1)$ th compressions [5]. Huang et al. [24] proposed a new method based on the changes in the DCT coefficient during single and double compressions of the videos to detect MPEG-2 double compression with the same bit rate. First, the number of dissimilar DCT coefficients between the single and double compressed videos were calculated and denoted as D' . Similarly, the number of dissimilar DCT coefficients between double and triple compressed videos were calculated and denoted as D'' . Then, they distinguished between the single and double compression of videos based on the below Equation:

$$\begin{cases} \text{Video is singly compressed if } D'' = D' \\ \text{Video is doubly compressed if } D'' > D' \end{cases} \quad (1)$$

Jiang et al. [25] extended the above idea in MPEG-2 and H.264/AVC by analyzing the quality reduction of the videos while performing multiple compressions using the same coding environments. In the MPEG-2 inter coding process, this method used macroblock information called macroblock mode (MBM), including the macroblock type $M_{type} \in \{\text{intra}_4 \times 4, \text{intra}_{16} \times 16, \text{P_Skip}, \text{inter}_8 \times 8, \text{inter}_{16} \times 16, \text{inter}_8 \times 16, \text{inter}_{16} \times 8\}$ and motion vector $M_{mv} \in \{M_{mv1}, M_{mv2}, \dots, M_{mv16}\}$ to determine the stability of the macroblocks at a fixed position between n th and $(n+1)$ th compressions for every P-frame. Then, the stability was calculated as follows:

$$\text{Stability} = \begin{cases} 1 & \text{if } \text{MBM}_1(M_{type}, M_{mv}) \neq \text{MBM}_2(M_{type}, M_{mv}) \\ 0 & \text{if } \text{MBM}_1(M_{type}, M_{mv}) = \text{MBM}_2(M_{type}, M_{mv}) \end{cases} \quad (2)$$

MPEG-2 inter coding strategy could also be used for the H.264/AVC inter coding process. The H.264/AVC intra coding process is different from the inter coding process. In the H.264/AVC intra coding process, macroblock types $M_{type} \in \{\text{intra}_4 \times 4, \text{intra}_{16} \times 16\}$ and macroblock prediction mode $M_{pre} \in \{\text{Vertical}, \text{Horizontal}, \text{DC}, \text{Plane}, \text{Diagonal-Down-Left}, \text{Diagonal-Down-Right}, \text{Vertical-Right}, \text{Horizontal-Down}, \text{Vertical-Left}, \text{Horizontal-UP}\}$ were used to calculate the stability as follows:

$$\text{Stability} = \begin{cases} 1 & \text{if } \text{MBM}_1(M_{type}, M_{pre}) \neq \text{MBM}_2(M_{type}, M_{pre}) \\ 0 & \text{if } \text{MBM}_1(M_{type}, M_{pre}) = \text{MBM}_2(M_{type}, M_{pre}) \end{cases} \quad (3)$$

These techniques were further extended to expose the video manipulation by detecting double compression in the HEVC compressed domain with the same coding parameters in intra coding modes using the picture partitioning information, including PU depth data. Jiang et al. [3] considered four PU blocks $\text{PU}_{type} \in \{4 \times 4, 8 \times 8, 16 \times 16, 32 \times 32\}$ and 35 prediction directions $\text{PU}_{pre} \in \{\alpha_i; i = 0, 1, \dots, 34\}$, which could be expressed similar to the H.264/AVC intra coding process. In [4], Elrowayati et al. observed that the quantized discrete sine transform coefficient of PB changes during multiple compressions. They integrated this concept with an intra prediction mode group (IPMG) to determine the similarity between neighboring blocks. The IPMG and the standard deviation of the autocorrelation feature between coefficients were combined to determine that two PBs belong to the same prediction block feature. This model can be formalized as follows.

$$\text{Stability} = \begin{cases} 1 & \text{if } \text{PBF}(\text{PB}_1) \neq \text{PBF}(\text{PB}_2) \\ 0 & \text{if } \text{PBF}(\text{PB}_1) = \text{PBF}(\text{PB}_2) \end{cases} \quad (4)$$

4. Proposed algorithm

4.1. Variation of statistical properties in partitioning information during recompressions

In the overall implementation of the system, we perform the analysis, preprocessing, and feature extraction based on the sequences that are compressed in the all intra (AI) and low-delay P (LDP) modes. We only consider I-frames including intra CU, intra PU, intra TU, and intra MPM information to detect HEVC double compression. This approach ensures that the system is fast and practical as we only consider independently coded I-frames in the AI and LDP modes. We first extract four types of information (CU, PU, TU, and MPM). Subsequently, we analyze the changes in information while the compression is performed. The changes of information in CU partitioning in single, double, and triple compressions of the first I-frame in the *Aspen* sequence are depicted in Fig. 4. As multiple compressions are conducted, the portions of 64×64 , 32×32 , 16×16 , and 8×8 CU depths change, as shown Fig. 4 by red squares. For PU information, we use the changes in the prediction modes while the compression is introduced several times. We consider four adjacent CTUs of the first I-frame in the *Aspen* sequence to visualize the changes as concrete evidence in the prediction modes. Fig. 5 shows the variability of the prediction mode in single, double, and triple compressions, as marked by the red squares.

4.2. Research motivation and major contributions

Studies from Section 3, many methods compared PU types, motion vectors, and prediction modes to determine the stability of each unit during multiple compressions. In some of them, a PU size of 32×32 or larger was not considered as the changes were very small during multiple compressions [3]. The classification was conducted using traditional machine learning techniques [26]. In this paper, we propose a new method for HEVC double compression detection with the same coding configurations using picture partitioning information, including the CU, PU, TU, and MPM based on deep learning.

Therefore, we propose two types of feature sets extracted from CU, PU, TU, and MPM information during multiple compressions under the same coding configurations. To detect HEVC double compression, we analyze the changes in picture partitioning properties by repetitive HEVC compressions and extract meaningful statistical and DCNN features. Finally, we use two sets of features for the detection scenarios (in separate or combined manner). A detailed description of the proposed methodology is presented below.

4.3. Overall structure of the proposed method

We extract 5-D statistical and 62-D DCNN features to distinguish between the single and double compressions in the HEVC-coded videos. To calculate the statistical or DCNN features, we first extract the CU, PU, TU, and MPM partitioning and prediction information from the input sequences. The partitioning and prediction information includes the position, depth, and prediction mode information of a specified block. Then, the input as a single or double compressed video is recompressed again without changing the coding environment. The CU, PU, TU, and MPM information is extracted again during the recompressions. To obtain numerical value of the partitioning information, we use the position, and depth of the quad-tree structure as shown in Fig. 1(b) and the prediction modes of a block to represent the depth map in the spatial format. From the extracted depth maps, the absolute difference map (ADM) is obtained by subtraction between n th and $(n+1)$ th compressions.

The proposed method includes two main steps. The first step is the spatial representation of the partitioning and prediction mode information, followed by feature extraction from the ADM obtained from CU, PU, TU, and MPM depth maps. In the second step, the extracted features

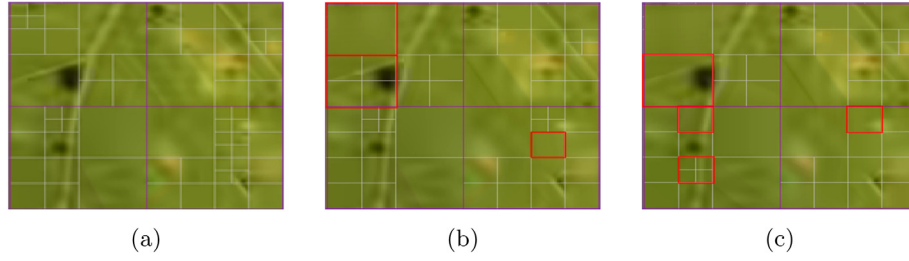


Fig. 4. CU partitioning in (a) single, (b) double, and (c) triple compressions, where red squares indicate the changes of partition information.

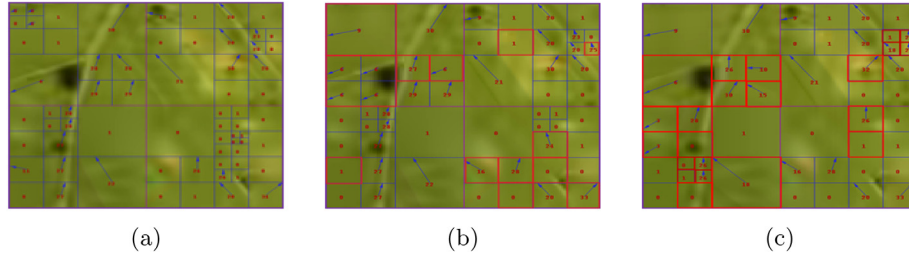


Fig. 5. PU prediction mode changes in (a) single, (b) double, and (c) triple compression, where red squares indicate changes of partition or prediction information.

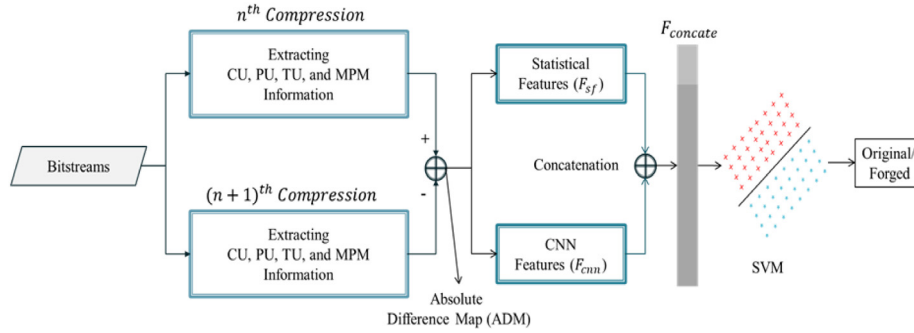


Fig. 6. Architecture of the proposed system.

are used for differentiating between single and double compressed videos in the HEVC compressed domain with the same coding configurations. Fig. 6 shows the overall structure of our proposed system. The input to the system is bitstream, which passes through the encoder and decoder for extracting the position, depth, and prediction modes of CU, PU, TU, and MPMs. After obtaining the desired information (position, depth, and prediction modes), we represent them in spatial formats and calculate the absolute difference between n^{th} and $(n+1)^{th}$ compressions for extracting statistical and DCNN features.

The next subsections detail the implementation of the proposed system by illustrating the spatial representation of the CU, PU, TU, and MPM depth map, feature extraction, and classifier specifications.

4.3.1. Spatial representation of CU, PU, and TU depth map

The recompressions introduce significant changes in the splitting and prediction mode information, as illustrated above in Section 4.1. We use these concepts to distinguish between single and double compressions. To the spatial representation of the CU and TU, we only consider location information such as position (height, width) and depth information. However, for PU, we consider location information such as position (height, width), depth, and prediction modes.

Let us consider a PB having block size d and prediction mode m . Thus, the PU depth mapping can be expressed using Eq. (5):

$$I_{PU}(x+i, y+j) = m; \quad i, j = 0, 1, \dots, d_{PU} - 1 \quad (5)$$

where I_{PU} is the depth map representation of PU, x and y represent the top-left position of PU; d_{PU} represents the size of PU, which is any one

of 32, 16, 8, or 4, and m indicates the 35 prediction modes. Therefore, the same values are assigned to the same PUs in I_{PU} .

The depth map of MPMs (I_{MPMs}) is represented in spatial formats for training the DCNN network by using an Equation similar to (5) except for three channels as follows:

$$I_{MPM}(x+i, y+j) = [MPM_1, MPM_2, MPM_3]; \quad i, j = 0, 1, \dots, d_{PU} - 1 \quad (6)$$

Similarly, the equation for the CU depth map of a coding block (CB) can be written as

$$I_{CU}(x+i, y+j) = n; \quad i, j = 0, 1, \dots, d_{CU} - 1 \quad (7)$$

where I_{CU} is the depth map representation of CU, n indicates the quad-tree depth ranging from 0 to 3, and d_{CU} is the size of CU (64, 32, 16, or 8). For the TU depth map, Eq. (7) is used to generate I_{TU} , but the only difference is the size of TU which is 32, 16, 8, or 4.

Fig. 7 shows the spatial representation of the CU, PU, TU, and MPM depth of the first I-frame in the *Aspen* sequence. After the spatial mapping of CU, PU, TU, and MPM, we compute the ADM between n^{th} and $(n+1)^{th}$ compressions. Let $I_1(x, y)$ and $I_2(x, y)$ be the single and double compressed I-frames, respectively, for one of I_{CU} , I_{PU} , I_{TU} , or I_{MPMs} . In this case, the ADM $I'(x, y)$ can be determined by Eq. (8):

$$I'(x, y) = |I_1(x, y) - I_2(x, y)|; \quad (8)$$

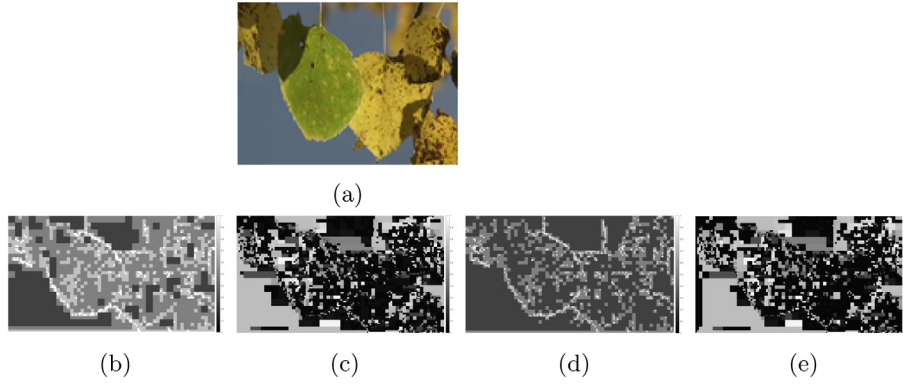


Fig. 7. Spatial representation; (a) original image, (b) CU, (c) PU, (d) TU, and (e) MPM depth maps.

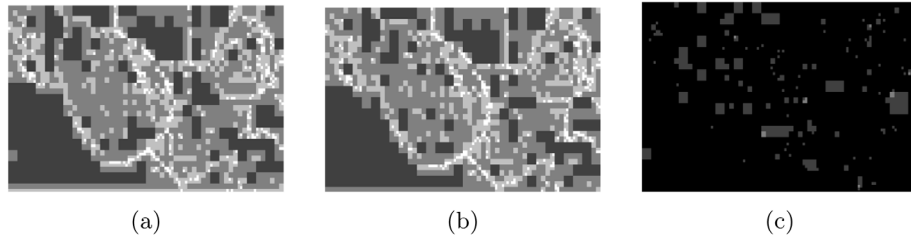


Fig. 8. ADM of CU depth; (a) single compression, (b) double compression, and (c) ADM.

Similarly, for double compression $I_2(x, y)$ and triple compression $I_3(x, y)$, the ADM $I''(x, y)$ can be defined as

$$I''(x, y) = |I_2(x, y) - I_3(x, y)|; \quad (9)$$

The ADM between single and double compression of the first I-frame in the *Aspen* sequence is shown in Fig. 8.

4.3.2. Feature extraction from ADM

We analyze the ADM obtained by the equations described in Section 4.3.1, and calculate some statistics of the histogram-based features such as mean, variance, skewness, kurtosis, and energy. Finally, we record the statistical features by concatenating all the above five features, as shown in Eq. (10), and feed them to a binary support vector machine (SVM) classifier.

$$\text{Statistical Features} : F_{sf} = [\text{Mean}, \text{Variance}, \text{Skewness}, \text{Kurtosis}, \text{Energy}] \quad (10)$$

The changes in these statistics for the ADM of the PU depth representation of 225 I-frames in the *Aspen* sequence are shown in Fig. 9.

We also consider a DCNN model for extracting the distinguishing features from the ADM obtained in the above sections of the single- and double-compressed videos. The input to the network is the ADM that passes through several layers. Nine blocks are used in the network inspired by [7]. In the first block, we use 8×8 non-overlapping convolutions with 16 output filters, batch normalization, and rectified linear unit (ReLU) followed by an average pooling. Because the spatial mapping of CU, PU, TU, and MPM is accomplished through the depth of size 64, 32, 16, 8, or 4, the non-overlapping even-sized convolution works better than the traditionally used odd-sized convolution. The blocks at positions 2nd to 4th and 6th to 8th are the same, which consist of convolutions with 3×3 kernel and 12 output filters, batch normalization, and ReLU. A transition layer is also inserted in the 5th position to reduce the network parameters shown in the orange box. The transition layer is composed of convolution, batch normalization,

and ReLU followed by the average pooling. The last block includes batch normalization, ReLU, and global average pooling for generating 62-D features. Finally, two fully connected layers are integrated to produce the final output for optimization purposes. A graphical representation of the proposed network is shown in Fig. 10.

We enumerate the gap between the predicted output and ground truth using a cross-entropy loss function, which performs significantly better in binary and multi-class classification. The general form of the cross-entropy loss for the target (y_i) and prediction (p_i) is defined as,

$$\text{loss} = - \sum_{i=1}^N y_i \log(p_i) \quad (11)$$

where N is the number of classes. In our case, N is equal to 2, which indicates the original and forged videos.

Finally, the 62-D DCNN features are extracted as

$$\text{DCNN Features} : F_{cnn} = \text{DCNN}_i; \quad i = 1, 2, \dots, 62 \quad (12)$$

The first and second features of the 62-D DCNN features for the PU depth map in the *Aspen* sequence are depicted in Fig. 11.

As described in the above section, we perform the concatenation of the statistical and the DCNN features of each depth map to create 67-D feature vector to obtain better results, which can be defined as,

$$\text{Concatenated Features} : F_{concat} = [F_{sf}, F_{cnn}] \quad (13)$$

5. Experimental results

5.1. Experimental details

Thirty-three video sequences of HD and thirty-three video sequences of wide video graphic array (WVGA) with YUV (4:2:0) format were used to evaluate the robustness of the proposed detection system. Each video had 225 frames [27–29]. We partitioned the dataset into training and testing sets. We used 17 videos for training the network and the remaining 16 videos for evaluation purposes to compare the results

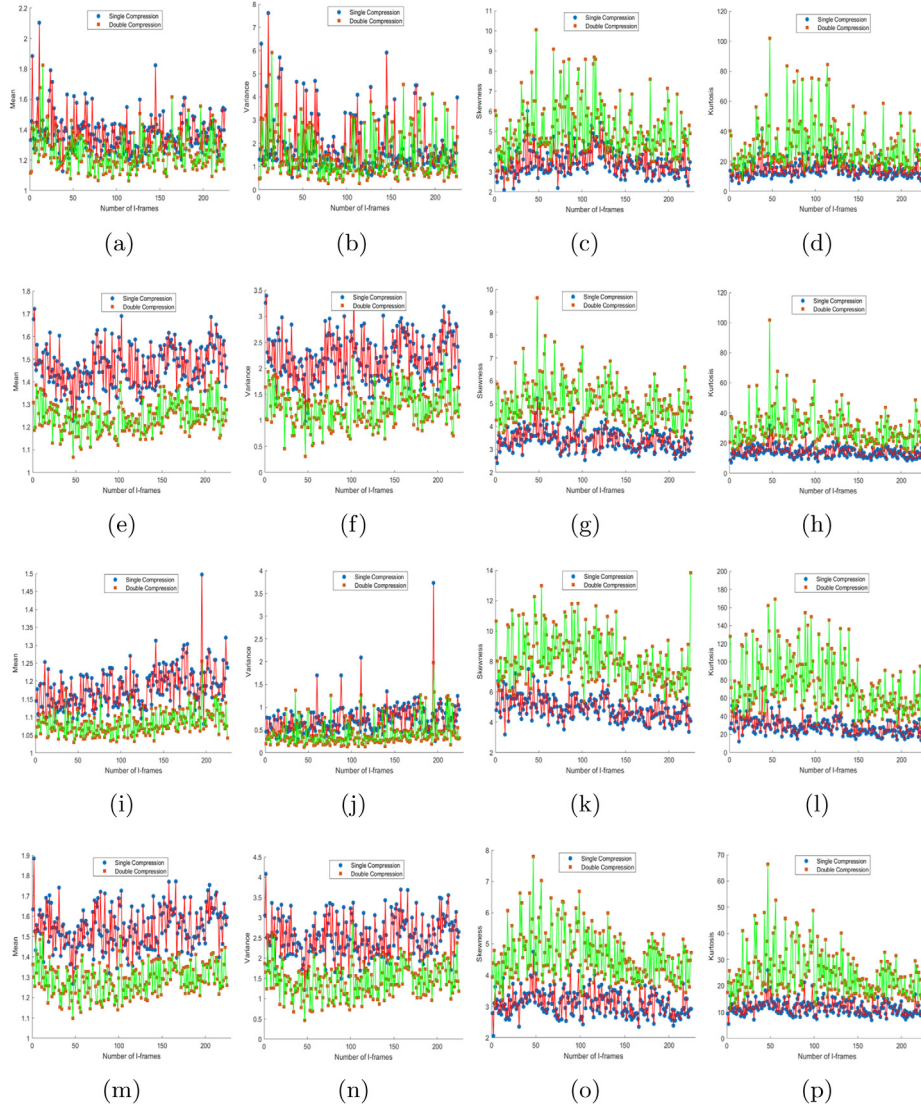


Fig. 9. Statistical features of ADM; (a) mean of CU, (b) variance of CU, (c) skewness of CU, (d) kurtosis of CU, (e) mean of PU, (f) variance of PU, (g) skewness of PU, (h) kurtosis of PU, (i) mean of TU, (j) variance of TU, (k) skewness of TU, (l) kurtosis of TU, (m) mean of MPM, (n) variance of MPM, (o) skewness of MPM, and (p) kurtosis of MPM.

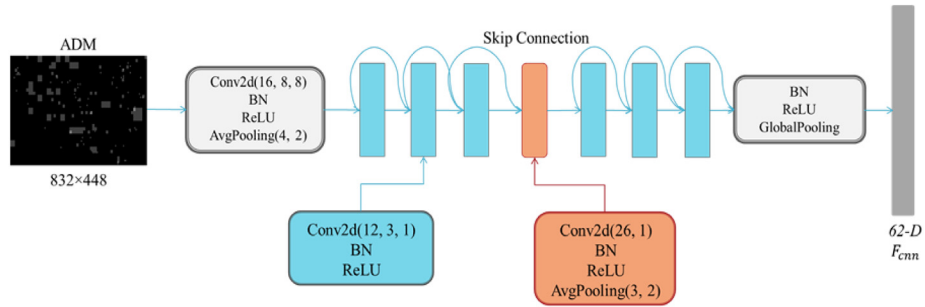


Fig. 10. DCNN network architecture.

using two subsets of features. The video sequences were compressed multiple times to extract the proposed features with the same coding configurations in the AI and LDP modes using the HM16.7, as the reference software [23]. The configuration details are listed in Table 1.

We performed three types of classification: (1) statistical features-based, (2) DCNN features-based, and (3) combination of statistical and DCNN features-based classification.

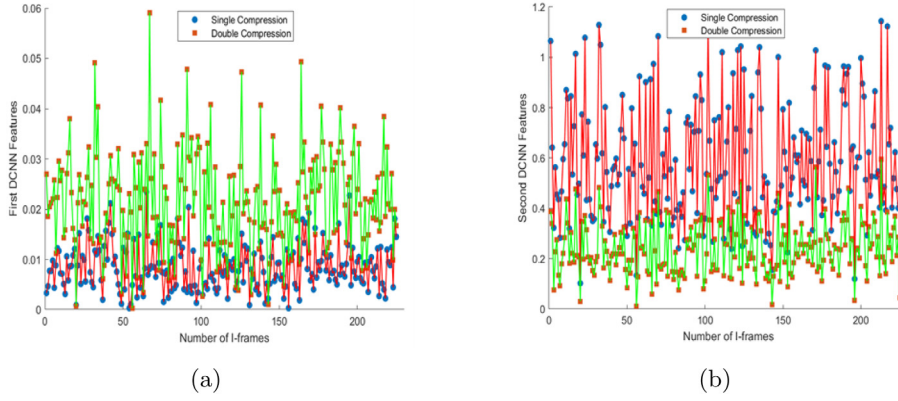


Fig. 11. DCNN features of PU's ADM; (a) first features and (b) second features.

Table 1

HEVC configurations.

HEVC CONFIGURATION PARAMETERS	VALUES
Modes	AI, LDP
Resolution of the sequences	832×448 (WVGA), 1920×1080 (HD)
Number of frames to be coded	225
QPs	22, 27, 32, 37
GOP Structure	IPPPPPPP

Table 2

Network configurations.

DCNN TRAINING PARAMETERS	VALUES
Number of epochs	100
Initial learning rate	0.001
Learning rate dropping factor	10%
Learning rate dropping period	Every 20th epoch

Accuracy is expressed by assuming the single compressed videos as negative observations (true negative, TN) and double-compressed videos as positive observations (true positive, TP). The following equation defines the accuracy in terms of TP and TN:

$$Accuracy(\%) = \frac{TP + TN}{Total\ Observation} \times 100 \quad (14)$$

where “total observations” is the total number of positive and negative samples used in the experiment.

5.2. Performance evaluations and comparisons

The training process of the DCNN network continued for 100 epochs in all the experiments. We initially set the learning rate as 0.001, which dropped 10% every 20th epoch. We set the mini-batch size of 32 (half positive and half negative samples). The momentum value was 0.999 in the Adam optimizer. Table 2 lists the details of the network configurations for training purposes.

The proposed network trains effectively: the training accuracies increase, and the losses decrease, as the number of epochs proceed. The detection results are evaluated in terms of accuracies for two different types of datasets on a subset of quantization parameters (QPs) $\in \{22, 27, 32, 37\}$ using CU, PU, TU, and MPM depth maps.

5.2.1. Experimental results in the AI mode

First, the experiment is conducted using the WVGA dataset with the statistical feature, the DCNN feature, and their combination in the AI mode. We present the results for each depth map separately to show the strength of the CU, PU, TU, and MPM depth maps for distinguishing between single and double compression. Table 3 shows the detection results for the WVGA dataset with a set of QPs for the statistical, DCNN, and concatenated features separately.

According to the results, PU and MPM are more useful than CU and TU. We consider only the position and depth information for CU and TU, whereas we integrated additional information (prediction modes) with the position and depth information for PU and MPM. The average detection accuracies of PU and MPM using DCNN features are 98.66% and 99.10%, respectively, which are considerably higher than the average accuracies of CU and TU (90.49% and 94.37%, respectively). We can also observe a similar trend for concatenated features: the average accuracies of PU and MPM are substantially better than those of the CU and TU depth maps. By assembling statistical and DCNN features, the proposed method achieves higher average accuracies for CU, PU, TU, and MPM depth maps.

Table 4 lists the experimental results for HD datasets. The PU and MPM are more robust for HD datasets (as for WVGA datasets). The proposed system accomplishes higher average accuracies of 99.22% and 99.39% for PU and MPM depth maps, respectively, whereas CU and TU depth maps attain 97.10% and 96.32%, respectively.

We have already ascertained that the proposed method accomplished the desired goal by individually applying each depth map (CU, PU, TU, and MPM). To obtain better results, we concatenate all the depth maps and evaluate the detection results for statistical, DCNN, and a combination of both statistical and DCNN features. We achieve a positive response in the performance improvements while assembling the features extracted from all the depth maps, as inferred in Table 5. We attained the highest average accuracies of 99.66% and 99.60% for the WVGA and HD videos, respectively, by gathering two types of features (statistical and DCNN features). Thus, the concatenation of the feature maps extracted from CU, PU, TU, and MPM depth maps enhances the classification capability of the system.

The method proposed by Jiang et al. [3] as described in Section 3, is the most modern and efficient state-of-the-art method for detecting HEVC double compression with the same coding information. They used SVM for classifying single and double compressions by combining the features extracted from PUs of sizes 4×4 , 8×8 , and 16×16 . For a larger feature set, SVM does not perform the classification well. Thus, we also conducted additional experiments using a neural network (NN) with three hidden layers having 30, 20, and 10 neurons to compare the results with our proposed system. Table 6 illustrates the results of the method suggested by Jiang et al. (using SVM and NN) and our proposed system. According to Table 6, the proposed method acquires the highest average accuracies of 99.66% and 99.60% for lower and higher resolution videos, respectively, which are approximately 3.5% and 1.7% greater than the method introduced by Jiang et al. using SVM. We attempted to distinguish between single and double compressed HEVC-coded videos by designing a neural network to apply in Jiang's method, which improved the performance by approximately 1% for WVGA and HD sequences. Even if we apply the NN in Jiang's method,

Table 3

Detection results of statistical, DCNN, and concatenated features generated from CU, PU, TU, and MPMs separately for WVGA sequences in AI mode.

QP	CU			PU			TU			MPM		
	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}
22	79.11%	87.21%	91.31%	87.97%	98.49%	99.05%	80.84%	90.32%	92.51%	91.65%	99.17%	99.43%
27	79.41%	90.92%	92.61%	86.49%	98.59%	98.94%	79.91%	94.03%	96.19%	90.16%	99.00%	99.32%
32	78.69%	91.19%	93.37%	86.83%	98.65%	99.10%	82.05%	96.50%	96.85%	89.88%	99.26%	99.35%
37	80.27%	92.64%	94.58%	87.09%	98.92%	99.27%	84.86%	96.64%	97.92%	90.14%	98.98%	99.29%
Avg.	79.37%	90.49%	92.97%	87.10%	98.66%	99.09%	81.92%	94.37%	95.87%	90.46%	99.10%	99.35%

Table 4

Detection results of statistical, DCNN, and concatenated features generated from CU, PU, TU, and MPMs separately for HD sequences in AI mode.

QP	CU			PU			TU			MPM		
	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}
22	87.53%	95.56%	97.21%	86.51%	98.18%	99.12%	79.17%	93.57%	96.08%	90.54%	98.78%	99.16%
27	87.07%	95.39%	97.30%	85.99%	98.31%	99.15%	79.31%	93.83%	95.53%	91.16%	98.83%	99.29%
32	86.78%	95.33%	96.92%	85.42%	99.03%	99.47%	79.95%	95.60%	96.69%	91.43%	99.11%	99.49%
37	86.92%	95.23%	96.95%	85.58%	98.58%	99.14%	80.44%	96.21%	96.99%	89.27%	98.85%	99.63%
Avg.	87.08%	95.38%	97.10%	85.88%	98.53%	99.22%	79.72%	94.80%	96.32%	90.60%	98.89%	99.39%

Table 5

Detection results of statistical, DCNN, and concatenated features generated from CU, PU, TU, and MPMs all together for WVGA and HD sequences in AI mode.

QP	WVGA			HD		
	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}
22	91.66%	99.33%	99.55%	93.58%	99.40%	99.87%
27	91.25%	99.61%	99.76%	92.39%	99.47%	99.61%
32	90.11%	99.47%	99.67%	92.16%	99.40%	99.56%
37	91.89%	99.69%	99.64%	91.15%	99.14%	99.36%
Avg.	91.23%	99.53%	99.66%	92.32%	99.35%	99.60%

Table 6

Performance comparisons in AI mode.

QP	Jiang et al. [3]				Proposed	
	WVGA		HD		WVGA	HD
	SVM	NN	SVM	NN		
22	94.33%	94.89%	96.93%	97.39%	99.55%	99.87%
27	95.46%	95.65%	97.86%	98.00%	99.76%	99.61%
32	97.40%	97.56%	98.06%	98.50%	99.67%	99.56%
37	97.54%	97.69%	98.69%	98.76%	99.64%	99.36%
Avg.	96.18%	96.45%	97.89%	98.16%	99.66%	99.60%

Table 7

Cross-QPs detection results with WVGA sequences.

QP	22	27	32	37
22	99.55%	82.42%	71.14%	64.59%
27	88.84%	99.76%	89.22%	83.25%
32	71.85%	90.37%	99.67%	86.89%
37	62.56%	80.42%	88.69%	99.64%

our system shows promising results, with approximately 3.2% and 1.5% increases in WVGA and HD sequences, respectively.

We have evaluated the cross-QP and cross dataset results for a better explanation of the strengths and limitations of the proposed method, as given in Tables 7–10. The left QPs in Tables 7 and 8 indicate the training, and the top QPs represent test cases. The detection accuracies for cross-QPs differ in descending order. It is because the partitioning information varies at a high rate from the current QP to its nearest and farthest QPs due to the quality degradation.

The detection accuracies for cross-dataset differs a lot because of the large quantity of partitioning information in HD and less in WVGA datasets. Thus, the proposed method depends on specific QPs and datasets for the detection of double compression.

To emphasize the robustness of the proposed system, we refined the effectiveness by prescribing the detection accuracy separately for

Table 8

Cross-QPs detection results with HD sequences.

QP	22	27	32	37
22	99.87%	86.25%	71.47%	64.57%
27	90.74%	99.61%	89.87%	73.51%
32	78.15%	92.39%	99.56%	85.34%
37	67.30%	84.23%	88.61%	99.36%

Table 9

Cross-dataset results (Training with HD sequences and testing with WVGA sequences).

QP	22	27	32	37
22	70.39%	83.07%	81.81%	83.61%
27	80.22%	83.77%	82.58%	77.86%
32	84.94%	79.64%	78.72%	70.83%
37	72.72%	77.95%	75.92%	81.34%

Table 10

Cross-dataset results (Training with WVGA sequences and testing with HD sequences).

QP	22	27	32	37
22	60.75%	58.98%	56.87%	59.84%
27	52.32%	61.56%	65.15%	71.60%
32	52.41%	60.87%	64.81%	76.44%
37	54.54%	66.97%	78.39%	81.91%

every test sequence. We considered 16 test sequences for showing the individual results using the same QP sets, as described earlier. By observing the detection results in Fig. 12(a) and (b) for WVGA and HD sequences, it can be seen that the resolved method fulfills the intended task with an average accuracy greater than 99.50% for all QPs.

5.2.2. Experimental results in the LDP mode

As more videos in real-world applications are encoded in the random access (RA) or LDP modes, we also consider the experiment in the LDP mode. As we use only I-frames, detection accuracy would be the same as in the LDP and RA modes. Tables 11 and 12 list the detection results for WVGA and HD datasets for CU, PU, TU, and MPM depths separately. Table 13 shows the concatenated results for all of the depth maps. The proposed method ensures approximately 99.46% and 99.33% detection accuracies for WVGA and HD datasets, respectively, which are almost identical to the AI mode.

We again investigate the performance of the proposed system with the prior method [3] given in Table 14. The proposed method shows

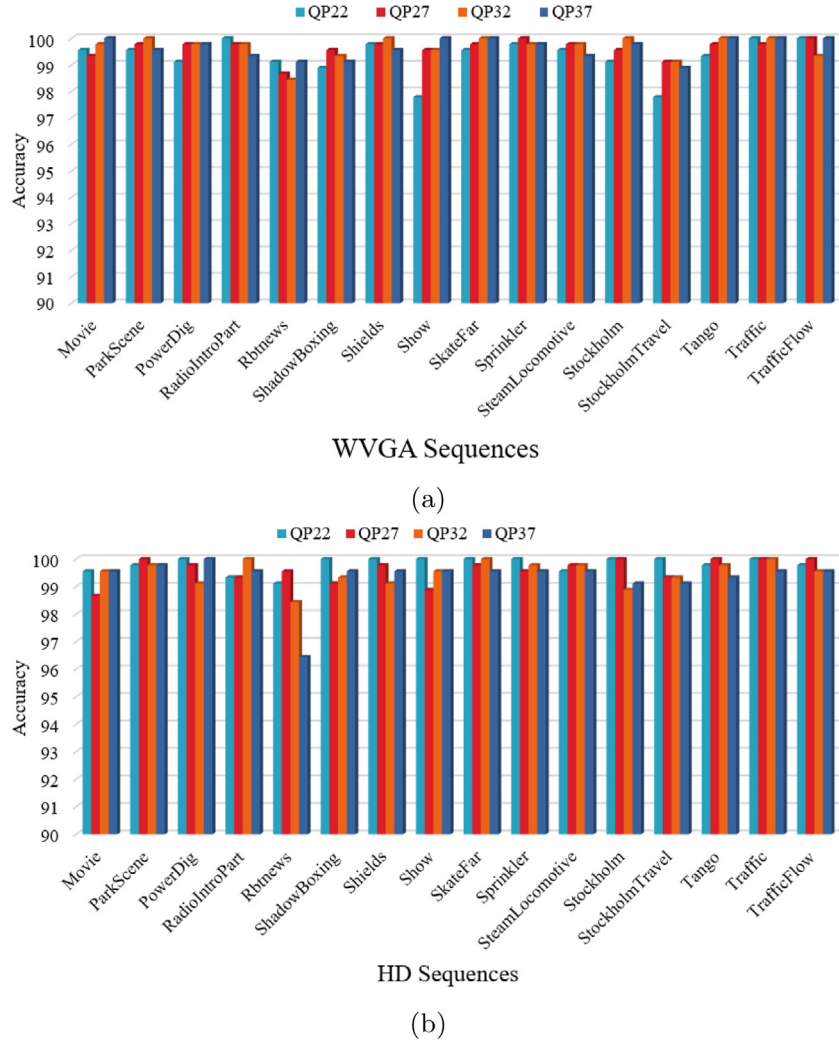


Fig. 12. Detection accuracy based on an individual sequence in AI mode; (a) WVGA and (b) HD sequences.

Table 11

Detection results of statistical, DCNN, and concatenated features generated from CU, PU, TU, and MPMS separately for WVGA sequences in LDP mode.

QP	CU			PU			TU			MPM		
	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}
22	80.82%	86.75%	88.90%	87.72%	98.23%	98.49%	82.54%	91.18%	95.41%	90.52%	99.03%	99.35%
27	79.31%	90.30%	93.75%	87.20%	98.04%	98.92%	83.94%	95.15%	95.47%	90.09%	98.49%	99.14%
32	78.71%	90.09%	92.46%	86.47%	98.06%	98.49%	82.76%	95.47%	95.58%	89.54%	98.28%	98.92%
37	78.88%	91.59%	93.43%	86.19%	98.69%	98.91%	85.02%	96.77%	97.41%	89.45%	99.03%	99.13%
Avg.	79.43%	89.68%	92.14%	86.90%	98.26%	98.70%	83.57%	94.64%	95.97%	89.90%	98.71%	99.14%

Table 12

Detection results of statistical, DCNN, and concatenated features generated from CU, PU, TU, and MPMS separately for HD sequences in LDP mode.

QP	CU			PU			TU			MPM		
	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}
22	85.42%	95.58%	96.55%	86.64%	98.17%	98.28%	79.66%	93.53%	95.04%	90.09%	98.49%	99.34%
27	83.89%	95.15%	97.90%	83.62%	98.06%	98.49%	80.60%	93.74%	94.29%	90.48%	98.60%	99.14%
32	82.06%	94.07%	96.23%	84.39%	98.81%	99.46%	80.84%	94.72%	96.01%	89.22%	98.28%	98.92%
37	83.62%	95.37%	95.47%	85.02%	98.17%	99.35%	82.56%	94.83%	96.55%	90.12%	98.17%	98.49%
Avg.	83.75%	95.04%	96.54%	84.92%	98.30%	98.90%	80.92%	94.21%	95.47%	89.98%	98.39%	98.97%

better performance in the LDP mode, about 3.5% and 1.5% more for SVM and 3.3% and 1.2% more for NN in WVGA and HD datasets, respectively.

We demonstrate the DCNN network complexities in terms of the total number of parameters, the number of floating-point operations

(FLOPs), and the time required to detect each frame using all three depth maps (CU, PU, TU, and MPM). The time and memory complexities are provided in Table 15 by averaging the time spent for detecting each QP. The number of parameters, FLOPs, and running time are expressed in millions, giga-flops, and seconds, respectively.

Table 13

Detection results of statistical, DCNN, and concatenated features generated from CU, PU, TU, and MPMs all together for WVGA and HD sequences in LDP mode.

QP	WVGA			HD		
	F_{sf}	F_{cnn}	F_{concat}	F_{sf}	F_{cnn}	F_{concat}
22	91.27%	98.60%	99.78%	92.78%	98.91%	99.68%
27	90.95%	98.92%	99.35%	91.16%	98.81%	99.35%
32	89.63%	99.57%	99.46%	89.28%	98.49%	99.03%
37	90.41%	98.81%	99.25%	89.68%	98.60%	99.25%
Avg.	90.57%	98.98%	99.46%	90.73%	98.70%	99.33%

Table 14

Performance comparisons in LDP mode.

QP	Jiang et al. [3]				Proposed	
	WVGA		HD		WVGA	HD
	SVM	NN	SVM	NN		
22	93.86%	94.50%	96.44%	97.41%	99.78%	99.68%
27	95.26%	95.47%	97.77%	97.63%	99.35%	99.35%
32	96.98%	97.01%	98.60%	98.71%	99.46%	99.03%
37	97.52%	97.41%	98.49%	98.60%	99.25%	99.25%
Avg.	95.91%	96.10%	97.83%	98.09%	99.46%	99.33%

Table 15

Network complexity.

Parameters (M)	WVGA		HD	
	FLOPs (G)	Time (s)	FLOPs (G)	Time (s)
0.0314	0.0585	0.0027	0.3166	0.0126

Table 16

Parameters at each block of the DCNN.

SL No.	Operations	Parameters	SL No.	Operations	Parameters	
1	Conv2d	6,160	6	Conv2d	2,820	
	BatchNorm2d	32		BatchNorm2d	52	
	ReLU	–		ReLU	–	
	AvgPool2d	–		–	–	
2	Conv2d	1,740	7	Conv2d	4,116	
	BatchNorm2d	24		BatchNorm2d	24	
	ReLU	–		ReLU	–	
3	Conv2d	3,036	8	Conv2d	5,412	
	BatchNorm2d	24		BatchNorm2d	24	
	ReLU	–		ReLU	–	
4	Conv2d	4,332	9	BatchNorm2d	124	
	BatchNorm2d	24		ReLU	–	
	ReLU	–		AdaptAvgPool2d	–	
5	Conv2d	1,378	10	Linear	2,016	
	BatchNorm2d	124				
	ReLU	–				
	AvgPool2d	–	11			Linear
Total parameters		31,428				

To better understand how weights are divided between fully-connected and convolutional layers, we have listed the number of parameters at each block of the DCNN in Table 16.

6. Conclusion

In this paper, we propose a new technique to detect HEVC double compression in the intra-prediction mode with the same coding information. We extracted two classes of features, the statistical and the DCNN features, extracted from the difference CU, PU, TU, and MPM depth maps during the multiple compressions under the same coding configurations. We used the frame partitioning and prediction mode information in HEVC to analyze the quality degradation and content loss of the video sequences during the recompressions. While conducting the recompressions, the partitioning information such as block size and

prediction modes were changed. After the analysis, we observed that the CU, PU, TU, and MPM depth maps exhibited distinctive behavior while performing subsequent compressions in HEVC. Thus, the depth information is considered as the distinguishing feature that detects HEVC double compression in the intra-prediction mode without changing any coding parameters. Experiments were conducted on HD and WVGA video sequences with various QP settings. We examined the detection results in the AI and LDP modes to emphasize the strength of the proposed method in real-world applications. We also conducted the cross-QP and cross-dataset experiments for better understanding. According to the experimental results, the proposed method shows promising results for detecting double HEVC compression with the same coding environments in the AI and LDP modes using WVGA and HD datasets. We further provided the sequence-wise detection results to make them more robust in practical scenarios. As we use the partitioning information of only I-frame, the detection results for the AI and LDP modes have the same results. The proposed method outperforms the state-of-the-art method. Hence, it can be used in forensic analysis to identify possible video manipulation.

CRedit authorship contribution statement

Kutub Uddin: Conceptualization, Methodology, Software, Writing. **Yoonmo Yang:** Methodology, Experiment. **Byung Tae Oh:** Methodology, Reviewing and editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This research was supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of ICT, South Korea (NRF-2019R1F1A1063229), and by the GRRC program of Gyeonggi Province, South Korea [2020-B02, Study on 3D Point Cloud Processing and Application Technology].

References

- [1] J.H. Hong, Y. Yang, B.T. Oh, Detection of frame deletion in HEVC-coded video in the compressed domain, *Digit. Investig.* (2019).
- [2] S. Milani, M. Fontani, P. Bestagini, M. Barni, A. Piva, M. Tagliasacchi, S. Tubaro, An overview on video forensics, *APSIPA Trans. Signal Inf. Process.* 1 (2012) 1–18.
- [3] X. Jiang, Q. Xu, T. Sun, B. Li, P. He, Detection of HEVC double compression with the same coding parameters based on analysis of intra coding quality degradation process, *IEEE Trans. Inf. Forensics Secur.* (2019).
- [4] A.A. Elrowayati, M.F.L. Abdullah, A.A. Manaf, A.S. Alfagi, Tampering detection of double-compression with the same quantization parameter in HEVC video streams, in: 7th IEEE International Conference on Control System, Computing and Engineering (ICCSCE), 2017, pp. 174–179.
- [5] F. Huang, J. Huang, Y.Q. Shi, Detecting double JPEG compression with the same quantization matrix, *IEEE Trans. Inf. Forensics Secur.* 5 (4) (2010) 848–856.
- [6] J. Yang, J. Xie, G. Zhu, S. Kwong, Y.Q. Shi, An effective method for detecting double JPEG compression with the same quantization matrix, *IEEE Trans. Inf. Forensics Secur.* 9 (11) (2014) 1933–1942.
- [7] X. Huang, S. Wang, G. Liu, Detecting double JPEG compression with same quantization matrix based on dense CNN feature, in: 25th IEEE International Conference on Image Processing (ICIP), 2018, pp. 3813–3817.
- [8] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, S. Tubaro, Aligned and non-aligned double JPEG detection using convolutional neural networks, *J. Vis. Commun. Image Represent.* 49 (2017) 153–163.
- [9] P. Peng, T. Sun, X. Jiang, K. Xu, B. Liu, Y.Q. Shi, Detection of double JPEG compression with the same quantization matrix based on convolutional neural networks, in: IEEE Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), 2018, pp. 717–721.

- [10] T. Sun, W. Wang, X. Jiang, Exposing video forgeries by detecting MPEG double compression, in: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2012, pp. 1389-1392.
- [11] J.A. Aghamaleki, A. Behrad, Detecting double compressed MPEG videos with the same quantization matrix and synchronized group of pictures structure, *J. Electron. Imaging* 27 (1) (2018) 013031.
- [12] X. Jiang, W. Wang, T. Sun, Y.Q. Shi, S. Wang, Detection of double compression in MPEG-4 videos based on Markov statistics, *IEEE Signal Process. Lett.* 20 (5) (2013) 447-450.
- [13] H. Liu, S. Li, S. Bian, Detecting frame deletion in H. 264 video, in: International Conference on Information Security Practice and Experience, Springer, 2014, pp. 262-270.
- [14] H. Yao, S. Song, C. Qin, Z. Tang, X. Liu, Detection of double-compressed H. 264/AVC video incorporating the features of the string of data bits and skip macroblocks, *Symmetry* 9 (12) (2017) 313.
- [15] D. Liao, R. Yang, H. Liu, J. Li, J. Huang, Double H. 264/AVC compression detection using quantized nonzero AC coefficients, in: Media Watermarking, Security, and Forensics III, Vol. 7880, International Society for Optics and Photonics, 2011, 78800Q.
- [16] Q. Xu, T. Sun, X. Jiang, Y. Dong, HEVC double compression detection based on SN-PUPM feature, in: International Workshop on Digital Watermarking, Springer, Cham, 2017, pp. 3-17.
- [17] Q. Li, R. Wang, D. Xu, Detection of double compression in HEVC videos based on TU size and quantized DCT coefficients, *IET Inf. Secur.* 13 (1) (2018) 1-6.
- [18] L. Yu, Y. Yang, Z. Li, Z. Zhang, G. Cao, HEVC double compression detection under different bitrates based on TU partition type, *EURASIP J. Image Video Process.* 19 (1) (2019) 67.
- [19] M. Huang, R. Wang, J. Xu, D. Xu, Q. Li, Detection of double compression for HEVC videos based on the co-occurrence matrix of DCT coefficients, in: International Workshop on Digital Watermarking, Vol. 9569, Springer, Cham, 2015, pp. 61-71.
- [20] Z.H. Li, R.S. Jia, Z.Z. Zhang, X.Y. Liang, J.W. Wang, Double HEVC compression detection with different bitrates based on co-occurrence matrix of PU types and DCT coefficients, in: ITM Web of Conferences, Vol. 12, EDP Sciences, 2017, p. 01020.
- [21] X. Jiang, P. He, T. Sun, R. Wang, Detection of double compressed HEVC videos using GOP-based PU type statistics, *IEEE Access* 7 (2019) 95352-95363.
- [22] I.K. Kim, J. Min, T. Lee, W.J. Han, J.H. Park, Block partitioning structure in the HEVC standard, *IEEE Trans. Circuits Syst. Video Technol.* 22 (12) (2012) 1697-1706.
- [23] G.J. Sullivan, J.R. Ohm, W.J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard, *IEEE Trans. Circuits Syst. Video Technol.* 22 (12) (2012) 1649-1668.
- [24] Z. Huang, F. Huang, J. Huang, Detection of double compression with the same bit rate in MPEG-2 videos, in: IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP), 2014, pp. 306-309.
- [25] X. Jiang, P. He, T. Sun, F. Xie, S. Wang, Detection of double compression with the same coding parameters based on quality degradation mechanism analysis, *IEEE Trans. Inf. Forensics Secur.* 13 (1) (2018) 170-185.
- [26] C.C. Chung, C.J. Lin, LIBSVM: A library for support vector machines, *ACM Trans. Intell. Syst. Technol. (TIST)* 2 (3) (2011) 1-27.
- [27] <https://www.cdvl.org/find-videos/>.
- [28] <https://media.xiph.org/video/derf/>.
- [29] http://ivc.univ-nantes.fr/en/databases/1080i_Videos/.