Detection and Localization of Video Transcoding From AVC to HEVC Based on Deep Representations of Decoded Frames and PU Maps

Haichao Yao, Rongrong Ni, Hadi Amirpour, Christian Timmerer, *Senior Member, IEEE*, Yao Zhao, *Senior Member, IEEE*

***Abstract*—In general, manipulated videos will eventually undergo recompression. Video transcoding will occur when the standard of recompression is different from the prior standard. Therefore, as a special sign of recompression, video transcoding can also be considered evidence of forgery in video forensics. In this paper, we focus on the detection and localization of video transcoding from AVC to HEVC (AVC-HEVC). There are two probable cases of AVC-HEVC transcoding — whole video transcoding and partial frame transcoding. However, the existing forensic methods only consider the detection of whole video transcoding, and they do not consider partial frame transcoding localization. In view of this, we propose a framewise scheme based on a convolutional neural network. First, we analyze that the essential difference between AVC-HEVC and HEVC is reflected in the high-frequency components of decoded frames. Then, the partition and location information of prediction units (PUs) are introduced to generate frame-level PU maps to make full use of the local artifacts of PUs. Finally, taking the decoded frames and PU maps as inputs, a dual-path network including specific convolutional modules and an adaptive fusion module is proposed. Through it, the artifacts on a single frame can be better extracted, and the transcoded frames can be detected and localized. Coupled with a simple voting strategy, the results of whole transcoding detection can be easily obtained. A large number of experiments are conducted to verify the performances. The results show that the proposed scheme outperforms or rivals the state-of-the-art methods in AVC-HEVC transcoding detection and localization.**

***Index Terms*—Video forensics, transcoded HEVC detection and localization, deep learning, dual-path network**

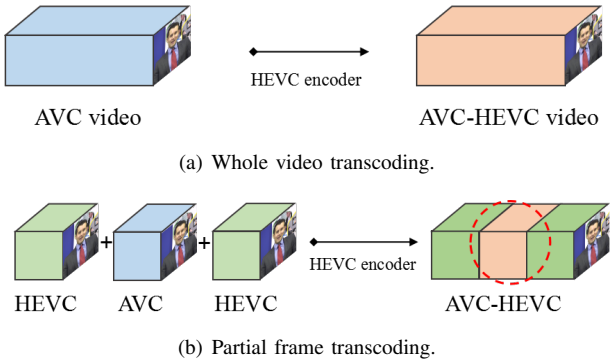
I. Introduction

CURRENTLY, computer technology and artificial intelligence are developing rapidly, and editing of digital images and videos is very common. Due to the existence of the advanced internet and extensive applications, the authenticity of images and videos is extremely significant. As the technologies to passively authenticate digital videos, many forensic works [1]–[4] have emerged.

This work was supported in part by the National Key R&D Program of China (No. 2021ZD0112100), National NSF of China (No. U1936212, No. 62120106009), and Beijing NSF (No. 4222014). *(Corresponding author: Rongrong Ni)*

H. Yao, R. Ni, Y. Zhao are with Institute of Information Science, Beijing Jiaotong University, Beijing 100044, China and also with Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China. (e-mail:17112062@bjtu.edu.cn, rrni@bjtu.edu.cn, yzhao@bjtu.edu.cn)

H. Amirpour and C. Timmerer are with the Christian Doppler Laboratory ATHENA, Institute of Information Technology, AlpenAdria-Universität Klagenfurt, 9020 Klagenfurt, Austria. (e-mail: hadi.amirpour@aau.at, Christian.Timmerer@aau.at)



(a) Whole video transcoding.

(b) Partial frame transcoding.

Fig. 1. Schematic diagram of transcoding from AVC to HEVC in two cases.

Video forgery can be roughly divided into intraframe forgery and interframe forgery for different purposes. However, both interframe and intraframe forgery will generally lead to recompression. As a consequence, recompression detection is an important step for video forensics when the forgery is unknown. In the literature, researchers have made great efforts toward recompression detection [5]–[17]. When the standard in recompression is inconsistent with the prior standard, transcoding occurs. As a case of video recompression, the transcoding can also be used for video forensics. Recently, transcoding detection has also attracted much attention.

As the successor of the former video coding standard Advanced Video Coding (AVC), High Efficiency Video Coding (HEVC) is 50% more efficient than AVC [18]. Namely, HEVC can maintain the same video quality at half the bitrate or double the video quality at the same bitrate. Therefore, some specific works [19], [20] for fast AVC-HEVC transcoding exist. However, to obtain more favorable compression and transmission for high-definition videos, forgers may reencode the forged AVC videos by using HEVC tools. There are two situations of video transcoding from AVC to HEVC: 1) The original AVC video is entirely reencoded into an HEVC video as shown in Fig. 1(a), this is called whole video transcoding. 2) To change the video semantics, the AVC and HEVC clips are merged and reencoded into a new HEVC video, which is called partial frame transcoding because only part of the video is transcoded, as shown in Fig. 1(b).

When AVC and HEVC clips have similar background

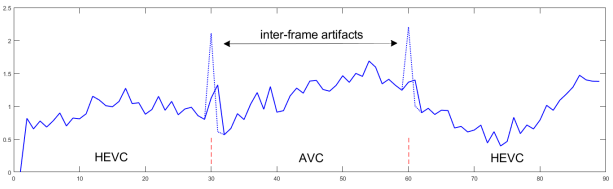


Fig. 2. Schematic diagram of interframe difference of a merged video before recompression. The abscissa represents frame index. The ordinate represents the average difference between two adjacent frames. The blue dotted line indicates the interframe artifacts brought by merging. And they can be easily repaired by interpolation of frames, as shown by the solid line.

and content, although their merging will bring interframe gaps, forgers can easily repair the gaps before recompression through interpolation so as not to leave interframe artifacts, as shown in Fig. 2. Therefore, partial transcoding detection and localization can be advisable means for forensics in this case. However, whether the video is transcoded entirely or partially is not known in blind detection. Although some methods [21]–[25] with fine performances in transcoding detection have been proposed, limitations still exist. They only consider the detection of whole video transcoding through overall statistical features but do not consider the transcoded frame localization, which requires local detection capabilities. In view of this, we propose a framewise approach that takes the frame as the minimum unit of detection and makes full use of the limited local and global information. Then, both whole transcoding detection and partial transcoding localization can be realized on the basis of framewise detection.

In this paper, we propose a dual-path convolutional neural network (CNN) to realize the framewise detection and localization of AVC-HEVC transcoding. One path uses PU maps generated by PU information on a single frame as inputs to learn the feature representation in the compressed domain. Another path employs a specially designed module with high-pass fixed convolutional kernels to learn the feature representation in the spatial domain. Then, an adaptive fusion module is designed to achieve the effective fusion of the dual-path features. Thus, the detection and localization of transcoded frames can be obtained. The detection results of whole video transcoding can also be easily obtained by using a simple voting strategy.

Our contributions are summarized as follows:

- Based on our theoretical analysis of the artifacts left in the spatial domain and compressed domain, we propose a dual-path network that represents a novel method for not only whole video transcoding detection but also the detection and localization of partially transcoded frames.
- We generate PU maps by introducing the location information. Then, the PU maps are used as the inputs of one path, which can make full use of the local information of PUs on a single frame. We employ high-pass fixed convolutional kernels to obtain the spatial high-frequency components as the inputs of another path. An adaptive fusion module is specially designed to take advantage of the two paths in feature learning.
- A large number of experiments are conducted to prove the effectiveness of our method for both transcoding detection and localization in treating different encoding parameters and video contents.

II. RELATED WORK

Recompression detection can be divided into two categories. On the one hand, the parameters used in recompression are consistent with the prior parameters. Huang *et al.* [5] proposed a double compression detection method for MPEG-2 video with the same bitrate based on the variation tendency of altered DCT coefficients in I-frames. Zhang *et al.* [6] proposed a ratio difference set to extract statistical features for multiple H.264/AVC compression detection with the same quantization parameters. Jiang *et al.* [7] proposed a double compression detection method for MPEG-x and H.264/AVC videos through quality degradation mechanism of multiple compression. Furthermore, Jiang *et al.* [8] analyzed the quality degradation mechanism and the source of error in HEVC videos, and proposed a method to detect double HEVC compression with the same coding parameters based on intra prediction mode.

On the other hand, the parameters used in recompression are different from the prior parameters. Wang *et al.* [9] proposed using static and temporal statistical perturbations caused by shifted group of picture (GOP) to detect MPEG-1 video double compression. Although shifted GOP may be caused by interframe forgery, which can be dealt with retrieval-based methods [26], [27], the change of GOP is very common in recompression. Vázquez-Padín *et al.* [10] proposed using variation of macroblock (MB) types in the P-frames to detect H.264/AVC video double compression. He *et al.* [11] proposed a method based on adaptive post-filtering to measure the intensity of block artifacts, and the detection of MPEG-4 double compression can be improved by combining the MB types artifacts. Recently, He *et al.* [12] utilized the encoding information in the compressed domain as the inputs, and proposed a hybrid network that combines CNN and long short-term memory to expose the artifacts of shifted GOP in recompressed HEVC videos. Yao *et al.* [13] discussed the detection of temporal traces in recompressed H.264/AVC video when adaptive GOP is enabled.

Another widely considered parameter is bitrate in recompression. To detect recompressed MPEG-2 videos with different bitrates from the prior bitrates, Su *et al.* [14] proposed using the specific convex pattern exhibited in the histogram of quantized DCT coefficients. Jiang *et al.* [15] proposed a Markov features based method to expose recompression with different quantization steps in MPEG-4 videos. Yu *et al.* [16] proposed detecting recompressed HEVC videos when the bitrate increases in recompression based on the prediction modes feature in the compressed domain. He *et al.* [17] designed a hybrid network that contains CNN structure to learn the features in the recompressed errors and multi-layer perceptron to learn the features in the histogram of zero-element clustered square regions.

Similar to GOP and bitrate, compression standards are optional to determine video quality, compression speed and storage size. Video transcoding will also introduce recompressed artifacts that can be used for video forensics. As for

the existing methods in AVC-HEVC transcoding detection, Costanzo *et al.* [21] observed that transcoding influences the motion prediction modes of subsequent HEVC compression in bi-directional predicted frames. However, this method is only applicable when the QPs in HEVC compression are smaller than those of AVC compression. Yu *et al.* [22] proposed a method based on the prediction unit (PU) statistical features in the first P-frame of all the GOPs. Besides, Bian *et al.* [23] proposed using the PU partition statistical features in I- and P-frames to realize AVC-HEVC detection. Moreover, Zhang *et al.* [24] proposed combining the coding unit (CU) and PU partition information of I- and P-frames to strengthen the statistical features. Recently, Xu *et al.* [25] combined the in-loop filtering features in interframes and PU statistical features in intraframes to expose transcoding. In a word, all the existing methods [22]–[25] extract video-level statistical features for whole video transcoding detection.

III. MODELING AND ANALYSIS

In this section, on the one hand, we introduce the theoretical analysis of the transcoding in the spatial domain to reveal the essential difference between transcoded and non-transcoded HEVC videos. On the other hand, we analyze the influence of transcoding in the compressed domain, especially the PU partition information. Then, the generation process of the PU map is presented.

A. Analysis in the spatial domain

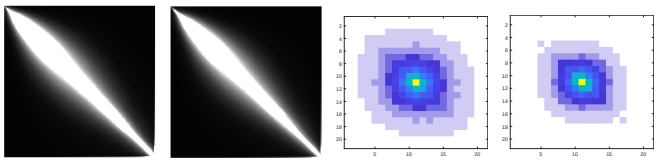
The compression process of a video is mainly composed of prediction, transformation, quantization and entropy coding. Suppose the original YUV sequence is V_{ori}, and the $i-th$ frame to be encoded is F_i. Therefore, the residual of the current frame can be obtained by $R_i = F_i - P_i$, where P_i represents reference pixels. Before being stacked into streaming, the residual R_i will undergo transformation and quantization as follows: $D_i = [\mathcal{T}(R_i)/Q_i]$, where $\mathcal{T}$ denotes the transformation such as DCT or DST, [.] denotes the rounding operation, Q_i and D_i denote the quantization matrix and quantized coefficients. Then, the compressed frame is decoded as $\hat{F}_i = \mathcal{L}[\mathcal{RT}(\mathcal{T}^{-1}(D_i \times Q_i) + P_i)]$, where $\mathcal{T}^{-1}(.)$ denotes the inverse transformation operation, $\mathcal{RT}(.)$ denotes the rounding and truncation operations, which will introduce value loss, and $\mathcal{L}[.]$ denotes in-loop filtering, which alleviates the blocking, ringing and other effects. Therefore, without losing generality, the change in the spatial domain brought by quantization, rounding, truncation, and in-loop filtering can be summed as ERR. Therefore, the relation between F_i, $\hat{F}_i$ and ERR can be represented as $\hat{F}_i = F_i + ERR_i$.

For clarity, we define the compression and decoding process as $\mathcal{C}(.)$. In this way, the decoded YUV after AVC compression is defined as $\mathcal{C}_{avc}(V_{ori}) = V_{ori} + ERR_{avc}$. Similarly,

$$\mathcal{C}_{hevc}(V_{ori}) = V_{ori} + ERR_{hevc} \tag{1}$$

and AVC-HEVC transcoded video can be formulated as follows:

$$\begin{aligned} \mathcal{C}_{hevc}(\mathcal{C}_{avc}(V_{ori})) &= \mathcal{C}_{hevc}(V_{ori} + ERR_{avc}) \\ &= V_{ori} + ERR_{avc} + ERR_{hevc} \end{aligned} \tag{2}$$



(a) HEVC. (b) AVC-HEVC. (c) HEVC. (d) AVC-HEVC.

Fig. 3. Statistical diagram of decoded frames. (a), (b) are the gray-level co-occurrence matrices of frames. (c), (d) are the gray-level co-occurrence matrices of frames after high-pass filtering.

In fact, ERR can be considered a high-frequency loss [28]. Specifically, ERR can be approximated as $N_{QRT} + N_{LP}$, where N_{QRT} denotes quantization, rounding and truncation loss, and N_{LP} denotes in-loop filtering loss. We mainly consider the quantization loss here because the loss of rounding and truncation is relatively slight. On the one hand, since some high-frequency coefficients are quantized to 0, N_{QRT} can be regarded as high-frequency loss caused by quantization. On the other hand, in-loop filtering counteracts the inconsistency and fluctuation caused by blocking and ringing effects, so N_{LP} can also be regarded as high-frequency loss.

With the help of new technologies such as quad-tree partitioning for precise prediction and sample adaptive offset filtering, HEVC performs much better in quantization and in-loop filtering operations than AVC, which will lead to the difference between high-frequency loss ERR_{avc} and ERR_{hevc}. In summary, comparing Formulas (1) and (2), we conclude that the essential artifacts of transcoding lie in the high-frequency components of decoded frames. To verify this, we use the decoded HEVC and AVC-HEVC frames for statistical analysis.

The transcoded and non-transcoded videos for experiments are used to conduct the statistical analysis here. The AVC encoder is JM, the HEVC encoder is HM, and the bitrate is 4 Mb/s. A total of 252 720P videos are used to generate HEVC and AVC-HEVC videos. Since the gray-level co-occurrence matrix (GLCM) [29] is an effective tool for spatial feature representation, we utilize the average GLCM to reflect the transcoded artifacts. GLCM is used to calculate the probability of a specific gray-level pair. In detail, for decoded frame I, the range of the gray-level values is first determined. For the horizontal ($\rightarrow$) co-occurrence, the conditional probability P of each gray-level pair is calculated as:

$$P(m,n) = (\alpha = m \ and \ \beta = n \mid \alpha \rightarrow \beta) \tag{3}$$

where (m,n) is the value of the pair and α and β are two gray-level values located with horizontal ($\rightarrow$) co-occurrence in I. In this way, the average probability matrix M can be obtained from four co-occurrence directions (horizontal$\rightarrow$, vertical$\uparrow$, diagonal$\searrow$, anti-diagonal$\nearrow$).

On the one hand, we directly calculate the GLCM on the decoded frames, as shown in Fig. 3(a) - Fig. 3(b). There is no obvious difference between the transcoded and non-transcoded videos, which shows that the artifacts are not reflected directly in the spatial domain. On the other hand, to show the GLCM of high-frequency information, 30 high-pass fixed filters in the spatial rich model [30] are used to process I to obtain the

high-frequency components. Three examples of the filters are as follows:

$$\begin{bmatrix} -1 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{bmatrix}$$

Then, the GLCM is shown in Fig. 3(c) - Fig. 3(d). It is obvious that the transcoded frames are easier to distinguish from the non-transcoded frames on high-frequency information, which confirms our analysis on transcoded artifacts, and this is of significance to the subsequent feature representation.

B. Analysis in the compressed domain and generation of PU maps

Although the spatial artifacts on decoded frames are expected to be distinguishable, the ERR_{hevc} introduced by HEVC recompression in AVC-HEVC videos may bring new interferences to the decoded frames. Therefore, it is better to mine other ERR_{avc} artifacts that are not in the spatial domain. Then, we turn our attention to the compressed domain.

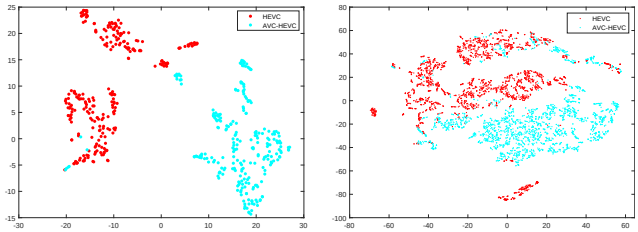
Compared with AVC, one of the most important improvements of HEVC is the flexible coding unit (CU). In HEVC, a frame is divided into several 64×64 nonoverlapping coding tree units (CTUs). A CTU can be directly used as a CU, or it can be further divided into several smaller CUs by the hierarchical quadtree-based technique. The smaller sizes include 32×32, 16×16, and 8×8. In this way, large CUs can improve the coding efficiency of the smooth areas, and small CUs can be used for details.

In the prediction of a CU, a prediction unit (PU) specifies all prediction modes of the CU. For a $2N \times 2N$ CU, there are two modes of intraframe PU: $2N \times 2N$ and $N \times N$, as well as eight modes of interframe PU: four symmetry modes ($2N \times 2N$, $2N \times N$, $N \times 2N$, and $N \times N$) and four asymmetric modes ($2N \times nU$, $2N \times nD$, $nL \times 2N$, and $nR \times 2N$), where n is equal to $N/2$. U, D, L, and R represent that the partition size of $N/2$ is in the upper, lower, left and right parts, respectively. The partitions of the CU and PU modes are all determined by rate-distortion optimization. Take the PU mode selection as an example. The optimization process can be described as selecting an optimal PU mode to minimize the total distortion of the CU when the total number of bits R is limited. Lagrange multipliers are used to obtain an unconstrained optimal function as follows:

$$min\ J, \quad J = \sum D_{i,j} + \lambda \sum R_{i,j} \tag{4}$$

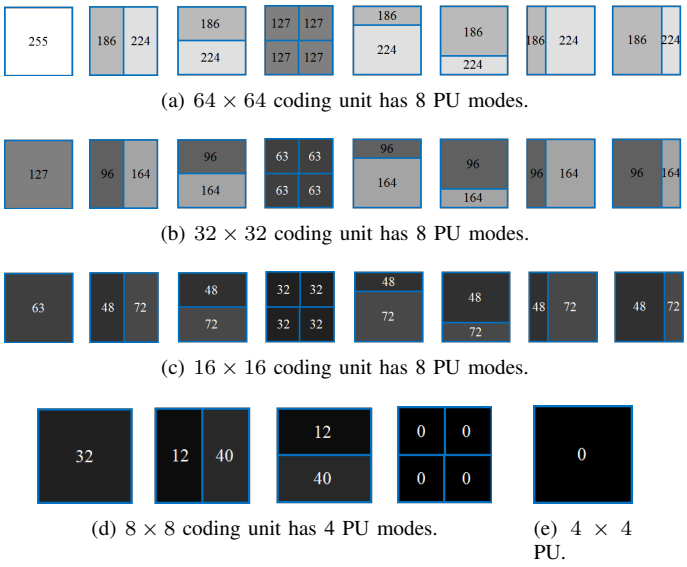
where $D(i,j)$ and $R(i,j)$ represent distortions and bits at position (i,j) in the CU. The encoder traverses all the PU partition modes and determines the optimal mode. In summary, the value and distribution of the pixels in the CU will directly affect the PU mode. Therefore, the PU partition information will contain the artifacts of transcoding indirectly.

As mentioned above, all the recent existing methods [22]–[25] used video-level statistical features of PU sizes for whole video transcoding detection. There are a total of twenty-five different PU sizes in the combination of 5 CU sizes and 8 PU modes in the interframes. Table I shows all twenty-five



(a) Video-level features (b) Frame-level features

Fig. 4. The t-SNE visualization of PU statistical features.



(a) 64×64 coding unit has 8 PU modes.

(b) 32×32 coding unit has 8 PU modes.

(c) 16×16 coding unit has 8 PU modes.

(d) 8×8 coding unit has 4 PU modes. (e) 4×4 PU.

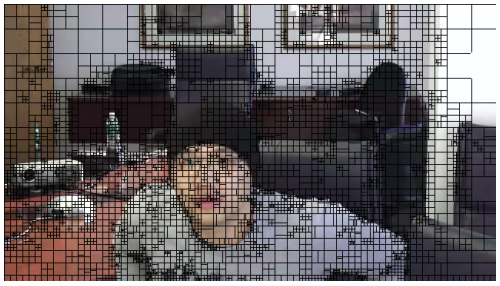
Fig. 5. Schematic diagram of grayscale values for different PU sizes. The number on the center of PU is the corresponding grayscale value.

different PU sizes. In partial frame transcoding detection and localization, however, the frame-level statistical features of PU sizes may not be sufficiently effective. Since the number of PUs on a single frame is limited, the statistical features can be greatly affected by the content. To confirm this, t-distributed stochastic neighbor embedding (t-SNE) is used for feature visualization. The same data as the statistical analysis in Sec. III-A are used, and the frame-level features of PU sizes are extracted. As shown in Fig. 4, the distribution of video-level features is conducive for detection. However, the frame-level features are so close that they may not be able to be distinguished.

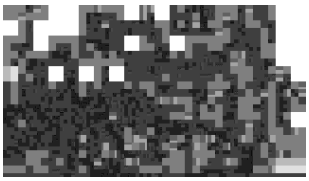
TABLE I
THE TWENTY-FIVE PU SIZES IN INTERFRAME.

order	size
1 ∼ 7	64×64, 64×32, 32×64, 64×16, 64×48, 16×64, 48×64
8 ∼ 14	32×32, 32×16, 16×32, 32×8, 32×24, 8×32, 24×32
15 ∼ 21	16×16, 16×8, 8×16, 16×4, 16×12, 4×16, 12×16
22 ∼ 25	8×8, 8×4, 4×8, 4×4

In view of this, more information needs to be mined on the PUs. Please note that statistical features only use the global statistics of PU sizes, and the local information of PUs is completely discarded. Since CNNs have strong local



(a) PU information in a 720P (1280×720 pixel) frame.



(b) The corresponding PU map (320×180 pixel).

Fig. 6. Schematic diagram of PU information and PU map we generated.

and global representation abilities, an end-to-end CNN-based solution is considered to be suitable for the current issue. To this end, the PU information needs to be processed so that it can be used as the input of the CNN. Therefore, a scheme is designed to convert the PU partition and position information of a frame into a grayscale image called the PU map.

For this purpose, the generated PU map needs to meet the following two requirements. 1) PUs of different sizes and modes should be reflected in different grayscale values. 2) The location of the PU should be reflected. Please note that a 4×4 PU in the interframe is the smallest unit, and all the other PU sizes are integral multiples of it. Therefore, the 4×4 PU can be represented by a pixel in the map with a specific value, an the $N \times M$ PU can be represented by $N/4 \times M/4$ pixels with different values.

We first consider the square PUs. There are five square PUs: 64×64, 32×32, 16×16, 8×8 and 4×4. To maximize the difference, we let the grayscale value of different PUs distribute in the range of 0∼255. As a result, the value of the largest 64×64 PU is set to 255, and the smallest 4×4 PU is set to 0. Then, the values of 32×32, 16×16, and 8×8 PU are set to 127, 64, and 32, respectively, by continuously halving.

Second, the values of the other rectangular PUs are determined. Since the $2N \times 2N$ CU is divided into two rectangular PUs, the values of the two rectangular PUs need to be different to reflect the corresponding modes and sizes. That is, 8 values (4 sizes×2 rectangular PUs) need to be set. Therefore, the details of PU maps with different sizes can be seen in Fig. 5. The number on the center of the PU is the corresponding value. We set 8 values (224, 186, 164, 96, 72, 48, 40, and 12) to the rectangular PUs because they need to have a certain distance from the existing five values (255, 127, 64, 32, and 0) and have a certain distance from each other.

Based on the above description, the PU information will be reflected as a PU map with $\frac{1}{16}$ of the original resolution, as shown in Fig. 6, which is beneficial for processing 720P or 1080P videos by the CNN. In this paper, we only consider the P-frames for feature representation because there are not many I-frames in the video, while B-frames do not exist in all the profiles. We use the PU maps of the P-frames as one of the inputs of the CNN to detect AVC-HEVC transcoding.

IV. Proposed method

Based on the theoretical analysis of the artifacts, we propose a dual-path network, as shown in Fig. 7. The network is composed of two paths — PU map path (PMP) and decoded frame path (DFP). They correspond to PU maps and decoded frames as inputs, respectively. In this section, we first provide the detailed descriptions of the two segments, artifact extraction segment and feature representation segment, in each path, and then the adaptive fusion segment is introduced. Finally, the voting strategy for video-level detection is introduced.

A. Artifacts extraction

In the artifact extraction segment, CNN components without pooling layers are used to learn the shallow features and prevent loss of artifacts. By introducing batch normalization (BN) and rectified linear unit (ReLU) in two convolutional layers, the chance of overfitting is alleviated, and nonlinear information is added. The combination described above is considered the basic convolutional (BasicConv) block in our proposed network, as shown in Fig. 7. In addition, instead of plain connection of BasicConv blocks, we employ shortcuts in both paths, which are more beneficial to feature learning and parameter convergence, as discussed in [31].

The PU maps are generated as reported in Sec. III-B, and the decoded frames are randomly cropped to the same size as the PU maps. The architecture that processes the inputs at first hand is also very important because its ability to extract artifacts will greatly affect the parameter learning of the following architectures. In steganalysis of digital images, Boroumand *et al.* [32] proposed using the basic convolutional layer. Namely, the best convolutional kernels should be learned, not hand-designed. However, in this transcoding detection problem, guided by our statistical analysis in Sec. III-A, high-pass fixed filters can provide distinguishable features as shown in Fig. 3. Therefore, similar to [33], [34], we employ 30 5×5 high-pass fixed filters in [30] with a stride of (1, 1) and padding of 0 as the convolutional layer to suppress the content of the decoded frames and obtain the beneficial high-frequency signals for BasicConv block learning in DFP.

In PMP, a basic convolutional layer with 2×2 kernels, stride of (1, 1) and padding of 0 is applied. The reason is that 8×4, 4×8, and 4×4 PUs will be finally represented as 2×1, 1×2, and 1×1 grayscale pixels on the PU map based on the PU map generation rule we designed. To target the small components in the PU map, we use a basic convolutional layer with 2×2 kernels which can extract artifacts between adjacent pixels (especially small PUs) more effectively and help to learn the local correlation within the PU map through the following BasicConv blocks.

The BasicConv blocks have the following parameters: 3×3 kernels, stride of (1, 1) and padding of 1. In this way, the

size of the feature map is invariant. As shown in Fig. 7, the numbers in the brackets represent the number of channels. In DFP, the artifact extraction segment aims to learn the useful feature map from the high-frequency signals, and a 30-channel feature map will be first generated after processing by 30 high-pass fixed kernels. We do not increase the channels of the feature map in the BasicConv blocks, so the artifact extraction segment finally outputs a 30-channel feature map. In PMP, to save GPU memory, the artifact extraction segment in PMP outputs a 16-channel feature map.

B. Feature representation

Next, the feature representation segment is designed to increase the number of channels, compact the size of the feature map and learn the final features in each path. It is composed of convolutional and pooling components, a convolutional layer with BN and a global average pooling (GAvgPool) layer. As shown in Fig. 7, pooling in the form of 3×3 averaging with a stride of (2, 2) and padding of 1 is applied behind BN, and we call the combination the convolution pooling (ConvPool) block. To preserve the shortcuts, we apply the form of 1×1 kernels, stride of (2, 2) and padding of 0 to the convolutional layer on the shortcuts. In this way, the feature maps processed by average pooling have the same size as the feature maps processed on the shortcuts, and elementwise summation can be applied to them. Therefore, after the ConvPool blocks, the size of the feature maps can be changed to $1/2^n$ of the input size, where n is the number of ConvPool blocks.

On the other hand, the channel of the feature map needs to be increased to obtain features. We propose gradually increasing the number of channels with a step in both paths. As shown in Fig. 7, the step is set to 32, and the feature representation segment in both paths involves 4 ConvPool blocks. Thus far, if 720P (1280×720) videos are processed by the proposed network, the size of inputs is 320×180 ($\frac{1}{16}$ of the frame resolution). After the artifact extraction segment and 4 ConvPool blocks, we can calculate that the sizes of the feature map are $128 \times 20 \times 12$ and $128 \times 20 \times 11$ in PMP and DFP. Finally, GAvgPool transforms them into 128-D features.

C. Adaptive fusion of dual-path features

In our designed dual-path network, there is no intersection before outputting the single-path features. This is because the information used by the two paths is relatively independent, and there is no correlation between two-path inputs that can be used for classification.

Since video compression involves many parameters, as well as the video contents, the features learned by the two paths may perform differently under different conditions. Therefore, a learned and effective fusion mechanism is needed to let more useful features play a more important role. As shown in Fig. 8, we design an adaptive fusion module that adopts a data-driven method to learn the weights of the two features and obtain the final fused features. Then, binary classification is performed through a fully connected layer.

More specifically, two Conv-ReLU-Conv-Sigmoid structures are designed for the representation of weights. In the first convolutional layer, 2D convolution is applied, so the 128-D features z are treated as a $128 \times 1 \times 1$ feature map. The reduction ratio r is introduced so that the number of channels in the middle feature map becomes C/r, which can reduce overfitting. Then, the channel number is increased with ratio r in the following convolutional layer. Thus far, the channelwise weights S before softmax can be formulated as

$$S = \mathcal{F}(\omega_2 \Phi(\omega_1 z)) \tag{5}$$

where $\mathcal{F}(\cdot)$ and $\Phi(\cdot)$ denote the sigmoid and ReLU functions, respectively. ω_1 and ω_2 are the weights of two hidden convolutional layers. To obtain the normalized weights, the softmax operation is applied on each dimension of the two learned weight vectors. Finally, the 128-D features are obtained by elementwise summation of two weighted features. Thus far, a dual-path network for frame-level transcoding detection has been established. Combined with a voting strategy on all the frames, the video-level detection results can be obtained.

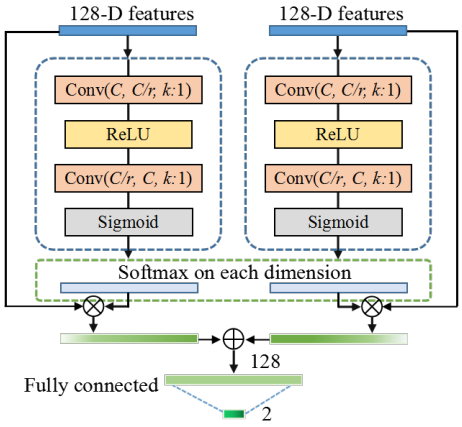


Fig. 8. Architecture of the adaptive fusion module. $\oplus$ and $\otimes$ represent elementwise summation and elementwise multiplication. Conv$(C, C/r, k:1)$ means the input of convolutional layer is C-channel feature map, the output is C/r-channel feature map, r is a reduction ratio, and k represents $k \times k$ kernels.

D. Voting strategy for the video-level detection

After obtaining the framewise classification results through the dual-path network, the video-level results are needed in the whole transcoding detection. Here, we utilize a simple majority voting strategy to obtain the video-level judgment of an N-frame video as follows:

$$\hat{m} = \arg\max_{k} \sum_{n=1}^{N} \Gamma(\nu_n == k) \tag{6}$$

where $\hat{m}$ indicates video-level detection result, $k \in 0, 1$ denotes negative or positive class, $\Gamma(\cdot)$ is the indicator function that takes 1 when the condition holds and 0 when the condition does not hold, $\nu_n \in 0, 1$ denotes the classification result of n-th frame, and '==' is to judge whether they are equal or not.

V. Experiments

This section investigates the performances of our proposed dual-path network in detecting transcoded videos and locating

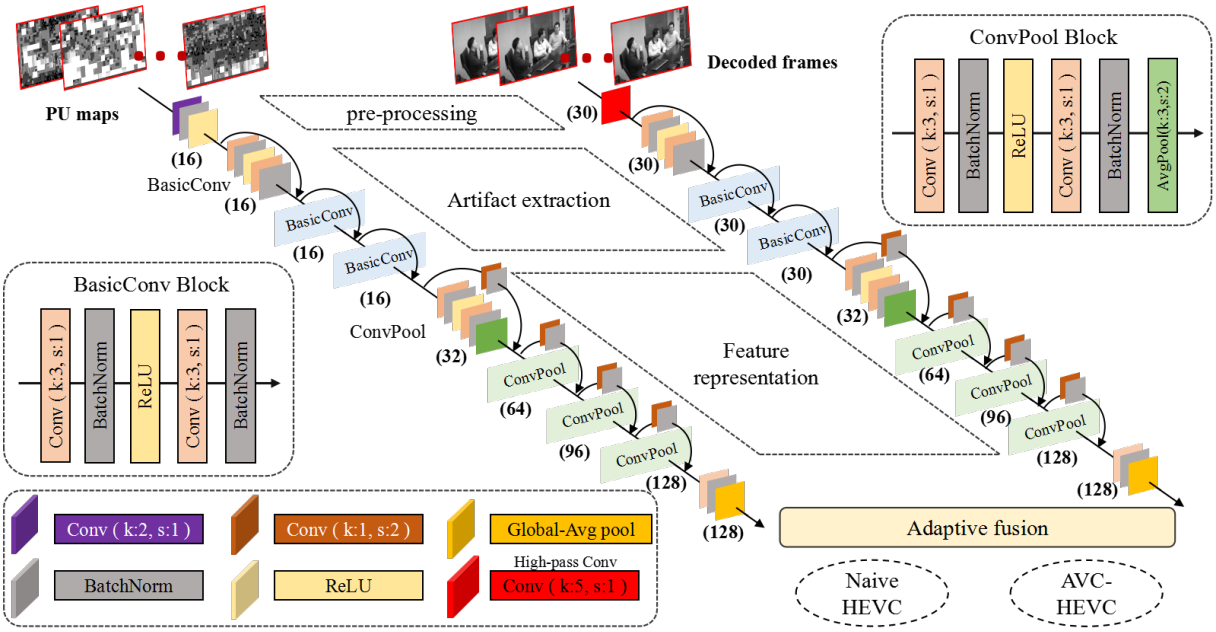


Fig. 7. Architecture of the proposed dual-path network for transcoding detection and localization. k and s denote kernal size and stride. The numbers in the brackets represent the number of channels.

the transcoded frames. In each subsection, the establishment of the datasets, the implementations, the setting of training and testing, the comparisons, and the analysis of the results are all reported in detail.

A. Detection of transcoded videos

TABLE II
THE SETTINGS OF DATASET FOR VIDEO TRANSCODING DETECTION. 'P' AND 'N' MEAN THE POSITIVE AND NEGATIVE SAMPLES. '(JM+HM)-HM' MEANS THAT THE AVC AND HEVC CLIPS ARE ENCODED BY JM AND HM, AND THEN THE MERGED YUV VIDEOS ARE ENCODED BY HM.

Transcoded video detection	
YUV videos	252 30-frames 720P YUV videos
Encoders	AVC: JM HEVC: HM and x265
Decoder	FFmpeg
Whole transcoding	P: 252 JM-HM videos, N: 252 HM videos P: 252 JM-x265 videos, N: 252 x265 videos
Partial transcoding	P: 77 (JM+HM)-HM videos, N: 77 HM videos P: 77 (JM+x265)-x265 videos, N: 77 x265 videos
Bitrates	4, 8, 15Mb/s
GOP structure	12, IPPP

1) Dataset: We use 14 720p YUV videos from [35] to establish the dataset. To increase the number of videos, each YUV video is divided into 30-frame nonoverlapping YUV clips. Therefore, a total of 252 720P YUV clips are generated as the experimental materials for the whole video transcoding detection.

For the choice of encoding tools, the official reference and frequently used software are both considered. JM [36] with the main profile is used for encoding AVC clips. HM [37] with low delay P main configuration and x265 [38] with *-amp* enabled slow preset are used for encoding HEVC clips.[1] FFmpeg [39] is used as the decoder for compressed videos.

The AVC-HEVC transcoded videos are considered positive samples, and the HEVC videos are considered negative samples. Two HEVC encoders — HM and x265 are considered. The details of the data for transcoding detection are presented in Table II. 'JM-HM' means the previous AVC, and the transcoded HEVC encoders are JM and HM. The corresponding negative samples are generated by HM. JM-x265 means that the AVC and HEVC encoders are JM and x265, respectively, and the encoder of the corresponding HEVC is x265. The most common coding mode, average bit rates (ABR), is chosen. Namely, the target bits are evenly distributed throughout the video in one-pass encoding. The encoding bitrates include 4, 8, and 15 Mb/s, which represent low, medium and high bitrate situations. The GOP (group of picture) size is set to 12. We consider the IPPP GOP structure because the existing methods and our proposed network mainly use information in the P-frames.

Furthermore, to generate the partially transcoded HEVC videos that may occur in video forgery, as shown in Fig. 1(b), the generated 30-frame AVC and HEVC clips are used to create new partially transcoded videos. As shown in Fig. 9, three 30-frame decoded YUV videos are merged (one of the three clips is AVC encoded and the other two are HEVC encoded) and then recompressed again via HEVC tools. In this way, only part of the successive frames are AVC-HEVC transcoded. To avoid losing generality, the transcoded frames appearing at the front, middle, and back of the sequence

[1]In the slow preset of x265 with *-amp* enabled, the rectangular partition $N \times 2N$ and $2N \times N$ and the asymmetric partition are enabled. In this case, the advances of HEVC in considering the quality and encoding speed can be reflected.

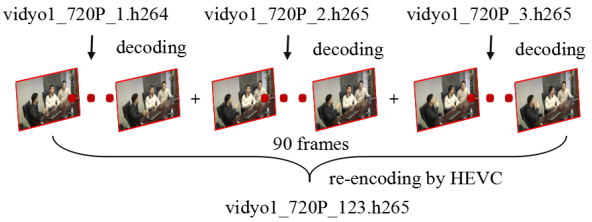


Fig. 9. Schematic diagram of the generation of partially transcoded videos.

have the same frequency. In that way, 231 AVC- or HEVC-encoded 30-frame clips are utilized to generate 77 90-frame partially transcoded HEVC videos, which can be expressed as '(JM+HM)-HM'. This means that the AVC and HEVC clips are encoded by JM and HM, and then the merged YUV videos are encoded by HM.

2) Comparisons and implementations: In the detection of video transcoding, three state-of-the-art (SOTA) methods [23]–[25] are implemented for comparison. They all extract the video-level statistical features, and then support vector machine (SVM) are used for classification. Based on their descriptions, we use the RBF kernel and Poly kernel to traverse the optimal gamma to obtain the classification results.

In our proposed method, not all the frames are fed to the network. This comes from the fact that the contents of adjacent frames are very similar. Therefore, one-third of the nonadjacent P-frames in each video are selected for frame-level training and testing. With the help of the voting strategy on frame-level results, the video-level results can be obtained. The dual-path network is implemented by PyTorch [40]. In the training process, stochastic gradient descent is applied as the optimization algorithm, and we set the momentum value as 0.9. The learning rate is initialized as 0.01 and multiplied by 0.5 every 5 epochs, and training ends after 30 epochs.

To demonstrate that the performances are independent of video content, all the experiments are performed in a cross-content manner. That is, all clips from the same YUV video only contribute to training or testing. The detection in this condition is closer to realistic scenarios. Moreover, to obtain more accurate and reasonable results, the training and test sets are randomly selected 3 times in all the evaluations (both the proposed method and SOTA). Namely, all the results reported in the tables are the average results of three trainings and tests. See the appendix for the division of training and test sets of different contents. Please note that we do not consider the cross-codec manner. For example, the models are trained on the HM data and tested on the x265 data. Because the artifacts left by two codecs are different. Even if the methods can learn valid features on the HM or x265 data, the features cannot be generalized to another codec. In fact, the encoder information and settings of video can be easily obtained by some software. So, we consider the training and testing under the same encoders and settings.

3) Criterion: To evaluate the performance, we use detection accuracy (Acc) as the criterion. It is the ratio of the number of positive and negative samples that are classified correctly to the total number of positive and negative samples and is defined as follows:

$$Acc = \frac{TP + TN}{P + N} \tag{7}$$

where P and N denote the number of positive and negative samples, respectively. TP and TN denote the number of positive and negative samples that are classified correctly. It is worth noting that the video-level Acc is used in detecting transcoded videos. However, our proposed network performs framewise detection, so the frame-level Acc is also used to reveal the performance of our method.

TABLE III
THE PERFORMANCES OF DIFFERENT NUMBER OF THE BASICCONV AND CONVPOOL BLOCKS. FRAME-LEVEL Acc ON 720P VIDEOS IN WHOLE VIDEO TRANSCODING DETECTION. ENCODERS ARE JM AND HM. THE UNDERLINED VALUE IS THE BEST RESULT.

BasicConv			ConvPool			4M	8M	15M
2	3	4	3	4	5			
	✓		✓			<u>0.9182</u>	0.9163	0.9064
	✓			✓		0.9117	<u>0.9395</u>	<u>0.9227</u>
	✓				✓	0.8871	0.9226	0.9026
✓				✓		0.9101	0.9282	0.9211
		✓		✓		0.9101	0.9393	0.9168

4) Verification of the settings: To determine the number of basic modules in the proposed network, experiments are first conducted. Since the features in the two paths need to have the same dimension to be processed in the adaptive fusion module, we keep the number of basic modules in the two paths consistent to control the dimension of features. The number of BasicConv and ConvPool blocks used in each path affects the artifact extraction and feature representation abilities, but too many blocks may lead to overfitting. As shown in Table III, the frame-level Acc in the whole video transcoding detection using different numbers of BasicConv and ConvPool blocks at 4, 8, 15 Mb/s bitrates is presented. The number of ConvPool affects the size of the feature map processed by Global-Avg pooling and the dimension of features. When the number of ConvPool is 4 (i.e., 128-D features) in each path, the dual-path network achieves better results. On the other hand, the number of BasicConv has little impact on the frame-level detection results, and the results of 3 BasicConv blocks are slightly better than others. In view of this, we choose 3 BasicConv and 4 ConvPool blocks in each path, as shown in Fig. 7, which can provide better results in a more balanced structure.

Next, we verify the setting of the first convolutional layer, which processes PU maps directly in PMP. From the results in Table IV, the 2×2 convolutional kernels achieve the best frame-level results compared to other sizes. Furthermore, the overall results when stripe=1 are better than those when stripe=2, which confirms our previous analysis in Sec. IV-A.

5) Whole video transcoding detection: The performances of whole transcoding detection are evaluated as shown in Table V. The 30-frame AVC-HEVC transcoded videos are regarded as positive samples, and the HEVC videos are regarded as negative samples. The HEVC compression maintains the same bitrates as the previous AVC. A single PMP and

TABLE IV
THE PERFORMANCES OF DIFFERENT SETTINGS OF INITIAL CONVOLUTIONAL LAYER IN PMP. FRAME-LEVEL Acc OF SINGLE PMP ON 720P VIDEOS IN WHOLE VIDEO TRANSCODING DETECTION. ENCODERS ARE JM AND HM. THE UNDERLINED VALUE IS THE BEST RESULT.

Frame-level Acc	4M	8M	15M
kernels:2×2, stride:(1,1)	0.9090	<u>0.9375</u>	<u>0.9043</u>
kernels:2×2, stride:(2,2)	<u>0.9159</u>	0.9169	0.9001
kernels:3×3, stride:(1,1)	0.9043	0.9288	0.9043
kernels:4×4, stride:(1,1)	0.9028	0.9232	0.8932
kernels:4×4, stride:(2,2)	0.9003	0.9035	0.9032

TABLE V
VIDEO-LEVEL PERFORMANCES OF WHOLE TRANSCODING DETECTION, THE AVC ENCODER IS JM, THE HEVC ENCODER IS HM OR X265. 'CONCAT' MEANS THE DUAL-PATH FEATURES ARE CONCATENATED. 'FUSION' MEANS THE FINAL FEATURES ARE OBTAINED THROUGH OUR PROPOSED ADAPTIVE FUSION MODULE.

Video-level Acc	AVC: JM, HEVC: HM			AVC: JM, HEVC: x265		
methods	4M	8M	15M	4M	8M	15M
Bian *et al.* [23]	0.9973	0.9934	0.9635	1.0000	0.9934	0.9765
Zhang *et al.* [24]	0.9947	0.9921	0.9632	1.0000	0.9882	0.9817
Xu *et al.* [25]	0.9700	0.9569	0.9583	0.9934	0.9830	0.9856
Single PMP	0.9726	0.9713	0.9388	0.9075	0.9296	0.9817
Single DFP	0.8775	0.9036	0.8749	0.9960	0.8828	0.8152
Dual-path concat	0.9452	0.9791	0.9375	0.9636	0.9752	0.9609
Dual-path fusion	0.9830	0.9791	0.9518	0.9947	0.9791	0.9752

DFP represent that only one path of our proposed network followed by a fully connected layer is applied for detection. As reported, although the SOTA methods perform slightly better in the whole video transcoding detection, our proposed method achieves comparable results. The reasons can be summarized as follows. All the SOTA methods are designed for the detection of whole transcoding, and they do not consider the localization of partial frames. Video-level features work well when one can assume that the video is either entirely transcoded or in the original format. However, in a blind scenario, it is not possible to know whether the video is entirely transcoded or only partially transcoded. To this end, we attempt to achieve frame-level detection. It is not only suitable for transcoded frame localization but can also realize the whole transcoding detection through the voting strategy. Although only the frame-level information is used, our proposed method still achieves valid performances similar to those of the SOTA methods. This also implies that our idea of frame-level detection with a voting strategy can be used for whole transcoding detection. In other words, the proposed method sacrifices a small amount of the whole transcoding detection accuracy in exchange for the ability of both detection and localization, which is valuable in blind scenarios.

In addition, the dual-path network has significantly better results than each single path, which shows the advantage of introducing both DFP and PMP. We also concatenate the dual-path features to obtain 256-D features. The fused 128-D features perform better, which shows the advantage of our designed adaptive fusion module. In summary, the results indicate that although our proposed method is motivated by solving partial transcoding detection and localization, it can also provide valid performances similar to video-level features in whole video transcoding detection.

TABLE VI
VIDEO-LEVEL PERFORMANCES OF PARTIAL TRANSCODING DETECTION, THE AVC ENCODER IS JM, THE HEVC ENCODER IS HM OR X265.

Video-level Acc	HM, 4M	HM, 8M	x265, 4M	x265, 8M
Bian *et al.* [23]	0.9572	0.7863	0.7778	0.8504
Zhang*et al.* [24]	0.9572	0.8546	0.8974	0.8760
Xu *et al.* [25]	0.8803	0.8418	0.9273	0.8846
Proposed	1.0000	1.0000	0.9871	0.9786

6) Partial transcoding detection: The performances of partially transcoded video detection are then evaluated, as shown in Table VI. The 90-frame partially transcoded videos as described in Sec. V-A1 are regarded as positive samples, the HEVC videos are regarded as negative samples, and all the methods are retrained. As one can observe, the performances of the SOTA methods are significantly reduced, especially at a relatively high bitrate. However, our proposed method still maintains high performance, even better than the results in whole transcoding detection. Please note that only one-third of successive frames in the partially transcoded video are AVC-HEVC frames. Namely, HEVC-HEVC frames occupy the majority of positive samples. Even though, the experimental results demonstrate that our proposed network can still classify the transcoded videos correctly, which means that AVC-HEVC and HEVC-HEVC frames can be distinguished from HEVC frames through our network. To sum up, the experiments on transcoded video detection show that our proposed method can provide fine performances in both whole transcoding and partial transcoding detection.

7) Ablation study of the dual-path network: For the ablation study of the dual-path network, on the one hand, we remove the high-pass filters and shortcuts, and then the frame-level Acc on each single path is compared. The experimental results are shown in Table VII. In DFP, it can be seen that the network is completely invalid when the high-pass fixed filters are removed, which demonstrates the necessity of high-pass fixed filters in this task. If the modules with shortcuts in PMP and DFP are replaced with plain modules, the frame-level classification results decrease. This indicates that plain structures are not capable of learning the features, while shortcuts can integrate more low-level information in this problem.

On the other hand, the frame-level Acc of the single-path and dual-path networks in whole transcoding detection is illustrated in Table VIII. 'Single PMP or DFP' represents a one-path network with only one kind of input: PU map or decoded frame, and 'Single PU&DF' represents a one-path network with the fused PU map and decoded frame as the input. As shown in Table VIII, the dual-path network performs better than each single-path network in most cases. Furthermore, when the encoder becomes x265, the influence of the coding

TABLE VIII
FRAME-LEVEL PERFORMANCES OF WHOLE TRANSCODING DETECTION, THE AVC ENCODER IS JM, THE HEVC ENCODER IS HM OR X265. 'SINGLE PU&DF' REPRESENTS A ONE-PATH NETWORK WITH FUSED PU MAPS AND DECODED FRAMES AS THE INPUTS.

frame-level Acc	AVC: JM, HEVC: HM			AVC: JM, HEVC: x265		
methods	4M	8M	15M	4M	8M	15M
Single PMP	0.9090	**0.9375**	0.9043	0.8199	0.8724	0.9267
Single DFP	0.8027	0.8093	0.8126	0.9301	0.7943	0.7590
Single PU&DF	0.8889	0.9201	0.9095	0.9326	0.9098	0.9419
Dual-path	**0.9117**	0.9329	**0.9157**	**0.9368**	**0.9249**	**0.9505**

parameters on the two paths and the role of the two paths are very clear. PMP does not perform very well at a low bitrate but improves with increasing bitrate. However, the performances of DFP are opposite to those of PMP. We summarize the reasons below. To improve coding efficiency, x265 does not use the most rate-distortion optimization in PU mode analysis [38]. In addition, when the bitrate is relatively lower, the exhaustive PU partition cannot be well conducted through the current rate-distortion level. This leads to a decrease in the function of PMP in this case. However, since the HEVC video can maintain much better high-frequency components than the AVC-HEVC transcoded video at low bitrates, the artifacts on DFP are more distinguishable. Then, DFP plays a more important role in the dual-path network. Therefore, we can conclude that the role of a dual-path network with DFP compared with a single PMP in whole transcoding detection lies in 1) improving the detection ability in most cases and 2) dealing with the situation in which PMP is not valid enough.

B. Localization of transcoded frames

In addition to transcoding detection, our proposed method can further locate the transcoded frames. In this subsection, more factors affecting partial transcoding localization are considered, and the performances are carefully verified.

1) Dataset: In contrast with transcoding detection, the essence of transcoded frame localization is the classification of AVC-HEVC and HEVC-HEVC frames. Besides the established 90-frame 720P data, to further explore the impacts of higher resolution and GOP structure in this task, we introduce 1080P data and IPBB GOP structure. Similar to the 720P data, 17 1080P YUV videos from [35] are first segmented into 30-frame nonoverlapping YUV clips. Then, 243 AVC or HEVC encoded clips are utilized to generate 81 90-frame partially transcoded HEVC videos. Since the encoding of 1080P videos under JM and HM is extremely time-consuming, and the default setting of JM needs to be modified to encode 1080P videos,[2] we choose x264 [41] as AVC encoder, and HM as HEVC encoder in 1080P videos with IPPP GOP.

The GOP size is 12 and we also consider the HEVC videos with IPBB structure that can be implemented by x265. To avoid losing generality, the transcoded frames appearing at the front, middle, and back of the sequence have the same frequency. We also use the one-pass ABR as coding mode. The details of the dataset for partial transcoding localization are presented in Table IX.

TABLE IX
THE SETTINGS OF DATASET FOR TRANSCODED FRAME LOCALIZATION. '(JM+HM)-HM' MEANS THAT THE AVC AND HEVC CLIPS ARE ENCODED BY JM AND HM, AND THEN THE MERGED YUV VIDEOS ARE ENCODED BY HM.

Transcoded frames localization	
IPPP GOP	720P, (JM+HM)-HM 720P, (JM+x265)-x265 1080P, (x264+HM)-HM
IPBB GOP	720P, (JM+x265)-x265 1080P, (x264+HM)-x265
Bitrates	4, 8Mb/s
GOP size	12

2) Comparisons and implementations: Since there is no specific method for partial transcoding localization, some features in SOTA methods that can be extracted locally are used for comparison. Therefore, the PU size features on P-frames used in [23], [24] are utilized as 25-D features. To further enhance the features, we also introduce the CU size features used in [24] to form combined CU&PU features. The P-frames contain four CU sizes: 64×64, 32×32, 16×16, and 8×8. In addition to the 25-D PU features, the combined 29-D CU&PU features extracted from P-frames are used for comparison. The reason why we do not use B-frames for localization is that B-frames do not exist in all the profiles of encoding, and the SOTA methods did not utilize the features related to B-frames. The AVC-HEVC transcoded frames are regarded as positive samples, and the HEVC-HEVC frames are regarded as negative samples. SVM with the experimental optimal parameters is used as the classifier for CU&PU features. We randomly select the training and test contents for three times, the specific division of training and testing is consistent with that in Sec. V-A2. All results are the average of three trainings and tests.

3) Criterion: Unlike transcoding detection, which focuses on the detection accuracy, we are also concerned about the false detection of positive and negative samples. In general, non-transcoded frames are in the majority, and false alarms will seriously affect the visualization of the localization. In view of this, $recall$ and false alarm (FA) are employed to reflect the overall classification performance. $recall$ is the ratio of the number of positive samples that are classified correctly to the total number of positive samples, FA is the ratio of the number of negative samples that are misclassified to the total number of negative samples, and the detailed definitions are as follows:

$$recall = \frac{TP}{TP + FN} \tag{8}$$

$$FA = \frac{FP}{TN + FP} \tag{9}$$

[2]The default setting of *numberreferenceframes* is 5 in the cfg file of JM. However, too many bits cause DPB (decoder picture buffer) overflow if reference frames are 1080P. Therefore, the default setting of *numberreferenceframes* must be reduced.

TABLE VII
THE ABLATION STUDY OF THE HIGH-PASS FILTERS AND SHORTCUTS IN THE PROPOSED NETWORK. FRAME-LEVEL Acc ON 720P VIDEOS IN WHOLE TRANSCODING DETECTION. ENCODERS ARE JM AND HM. 'PLAIN' MEANS NO RESIDUAL SHORTCUT. THE UNDERLINED VALUE IS THE BEST RESULT.

Decoded frames	PU maps	high-pass filters	plain BasicConv	shortcut BasicConv	plain ConvPool	shortcut ConvPool	4M	8M	15M
✓				✓		✓	0.5297	0.5426	0.5477
✓		✓	✓		✓		0.7731	0.7975	0.7964
✓		✓		✓		✓	0.8027	0.8093	0.8126
	✓		✓		✓		0.8986	0.9157	0.8971
	✓			✓		✓	0.9090	0.9375	0.9043

where TP and TN denote the number of positive and negative samples that are classified correctly, FP and FN denote the number of positive and negative samples that are misclassified.

In addition, to quantitatively reflect the localization effect in a single video, we define frame intersection over union ($FIoU$) which can be simply described as:

$$FIoU = \frac{\mathcal{F} \cap \mathcal{F}^{gt}}{\mathcal{F} \cup \mathcal{F}^{gt}} \tag{10}$$

where $\mathcal{F}$ denotes the frames that are classified as transcoded in a single video, and $\mathcal{F}^{gt}$ denotes the real transcoded frames. $\cap$ represents the number of overlapping frames, and $\cup$ represents the number of frames in the union.

4) Framewise localization: For the proposed dual-path network, the hyperparameters for training are identical to those in Sec. V-A2. The CU&PU features are extracted from a single frame. In fact, there are few positive samples and more negative samples. This is in line with the real-world situation, but is adverse to training, especially the proposed data-driven network. To address this problem, we eliminate the negative samples that are located in the first 5 frames and the last 5 frames of the 90-frame video in the training stage. Both CU&PU features and our method are implemented in this way.

The overall classification performances in the localization task are shown in Table X and Table XI. The AVC or HEVC clips and the corresponding partially transcoded HEVC videos have invariable bitrates. As one can observe from Table X, the overall performance of the proposed single PMP is much better than that of the CU&PU features. Moreover, to highlight the discriminative regions in the PU map, the class activation mapping (CAM) [42] is used for illustration. As shown in Fig. 10, the more active regions (in red) are distributed in the local parts of the background on the PU maps. This indicates that the transcoding artifacts in the PUs are locally distributed. Since the spatial position of the PUs is fully considered in our designed PU map, it is more conducive for our proposed network to learn the effective features on the PUs. In contrast with a single PMP, the proposed dual-path network has a smaller FA in most cases, although the $recall$ is also reduced, it is more favorable for visual localization.

Since the CU&PU features and the proposed method are to process P-frames, it is important to be effective when B-frames exist in the video. In view of this, we utilize the x265 IPBB structure to compress the AVC and HEVC merged YUV clips, and then only the P-frames in the transcoded HEVC videos are used for framewise training and testing. The classification results are shown in Table XI. One can observe that the classification becomes more challenging under the IPBB structure, especially in 1080P videos. When the resolution becomes larger, the artifacts are harder to learn from the spatial domain and PU map. Even so, our method is still improved relative to the CU&PU features to a certain extent.

TABLE XI
THE OVERALL CLASSIFICATION RESULTS OF TRANSCODED AND NON-TRANSCODED FRAMES IN TRANSCODING LOCALIZATION. THE GOP STRUCTURE IS IPBB.

	720P, 4M		720P, 8M		1080P, 4M		1080P, 8M	
Criteria	$recall$	FA	$recall$	FA	$recall$	FA	$recall$	FA
CU&PU	0.6954	0.0762	0.7325	0.0721	0.5867	0.2014	0.5283	0.2087
Dual-path	0.7506	0.0864	0.7637	0.0431	0.6189	0.1589	0.5500	0.1358

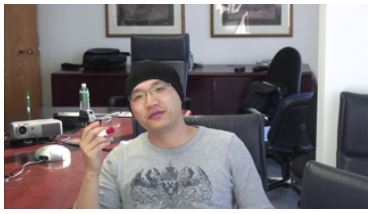
TABLE XII
THE AVERAGE $FIoU$ IN TRANSCODING LOCALIZATION.

average $FIoU$		CU&PU	PMP	Dual-path
IPPP	720P, (JM+HM)-HM, 4M	0.4811	0.6739	0.6506
	720P, (JM+HM)-HM, 8M	0.5144	0.7201	0.6950
	720P, (JM+x265)-x265, 4M	0.3728	0.5587	0.5773
	720P, (JM+x265)-x265, 8M	0.4178	0.6409	0.6401
	1080P, (x264+HM)-HM, 4M	0.4385	0.5564	0.5721
	1080P, (x264+HM)-HM, 8M	0.4013	0.5093	0.5109
IPBB	720P, (JM+x265)-x265, 4M	0.6047	0.6610	0.6763
	720P, (JM+x265)-x265, 8M	0.6392	0.7260	0.7528
	1080P, (x264+HM)-x265, 4M	0.3737	0.4113	0.4037
	1080P, (x264+HM)-x265, 8M	0.3231	0.4149	0.4182

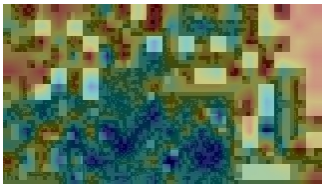
To further obtain the localization performances of a single video, the average $FIoU$ is used for the subsequent evaluation. First of all, since video forgery is often the manipulation of successive frames, it is reasonable to filter some separate false alarms in the videos. Second, since B-frames are not used for detection in our proposed method or the CU&PU features, the results of B-frames need to be represented by the adjacent P-frames in IPBB videos. Therefore, in practice, we perform the following postprocessing operations before calculating the $FIoU$: 1) The classified sequences of '00100' and '001100' are replaced by '00000' and '000000' in IPPP videos, where '0' represents a frame judged as a negative sample and '1' represents a frame judged as a positive sample. 2) The classification results of B-frames are represented by the nearest P-frame in front of them. After that, the average $FIoU$ of a single video is calculated as shown in Table XII.

TABLE X
THE OVERALL CLASSIFICATION RESULTS OF TRANSCODED AND NON-TRANSCODED FRAMES IN TRANSCODING LOCALIZATION. THE GOP STRUCTURE IS IPPP.

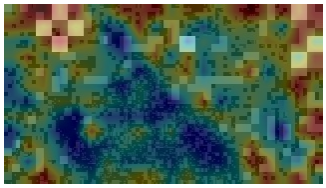
	720P, HM, 4M		720P, HM, 8M		720P, x265, 4M		720P, x265, 8M		1080P, HM, 4M		1080P, HM, 8M	
Criteria	$recall$	FA	$recall$	FA	$recall$	FA	$recall$	FA	$recall$	FA	$recall$	FA
CU&PU features	0.6763	0.0613	0.7196	0.0887	0.5800	0.1579	0.6829	0.1872	0.7009	0.1256	0.6520	0.1368
Single PMP	0.8771	0.1207	0.8723	0.1274	0.7189	0.1544	0.7789	0.1676	0.7669	0.1201	0.7022	0.1466
Dual-path	0.8320	0.0839	0.8218	0.0769	0.7272	0.1468	0.7175	0.1192	0.7485	0.0825	0.7007	0.1201



(a) *vidyo4*



(b) HEVC-HEVC PU map



(c) AVC-HEVC PU map

Fig. 10. CAM heatmaps in *vidyo4*. (b) and (c) are the CAM heatmaps of HEVC-HEVC PU map and AVC-HEVC PU map, respectively.

TABLE XIII
THE AVERAGE $FIoU$ OF SMALL WINDOW BASED TRANSCODING LOCALIZATION. 'WIN-SIZE' DENOTES THE SIZE OF THE WINDOWS.

Win-size	2	3	5	8	2	3	5	8
$FIoU$	720P, HM, 4M				720P, HM, 8M			
CU&PU	0.6322	0.6677	0.6978	0.6250	0.6384	0.6973	0.7099	**0.6731**
CU&PU&ILF	0.6070	0.6164	0.6804	0.6132	0.6313	0.6567	0.7166	0.6124
Proposed	**0.6996**	**0.6759**	**0.7063**	**0.6304**	**0.7444**	**0.7118**	**0.7449**	0.6457
$FIoU$	720P, x265, 4M				720P, x265, 8M			
CU&PU	0.4561	0.4864	0.6311	0.5986	0.4929	0.5386	0.6230	0.5683
CU&PU&ILF	0.5261	0.5143	0.6199	**0.6085**	0.5313	0.5708	0.6243	0.5999
Proposed	**0.6302**	**0.6155**	**0.6580**	0.5922	**0.6401**	**0.6936**	**0.6675**	**0.6623**
$FIoU$	1080P, HM, 4M				1080P, HM, 8M			
CU&PU	0.6304	0.6428	**0.6823**	**0.5745**	0.5473	**0.5560**	0.5557	0.5134
CU&PU&ILF	0.5797	0.5812	0.5971	0.5429	0.5056	0.5119	0.5444	0.4642
Proposed	**0.6313**	**0.6445**	0.6424	0.5450	**0.5496**	0.5450	**0.5625**	**0.5219**

It is obvious that both the single PMP and dual-path network perform better than the CU&PU features in all cases. Note that the results of a single PMP are slightly better than those of a dual-path network in some cases. This is because the higher $recall$ of a single PMP is beneficial for calculating the $FIoU$.

5) Window-based localization: Since meaningful forgery contains consecutive frames, the localization can be conducted in a window-like fashion. That is, the judgement of a window is used as the framewise results of all frames in the window. In fact, a larger size is not better because the windows may contain both non-transcoded and transcoded frames. In this way, regardless of whether the window is judged as a positive or negative sample, some frames are bound to be misjudged. Since the transcoded (forged) parts are randomly located in the video, the probability of including both non-transcoded and transcoded frames in a window increases with increasing size. In addition, if the window decreases, there will be less information available in the window, and the performances of statistical features are expected to be limited.

In fact, the SOTA methods [23]–[25] are not specifically designed for framewise or window-based localization. They all utilize the I-frame features. Therefore, only when the window is large enough can the SOTA methods be conducted. In view of the experimental data for localization, we randomly select 13, 18, 20, and 25 consecutive frames as the large windows to evaluate the methods in a window-based fashion. Please note that since the transcoded part is 30 consecutive frames in each video and the locations are relatively fixed at the front, middle and rear of the video, if the large size only leads to no window containing both non-transcoded and transcoded frames, it is too ideal to reflect the real-world situation. Our selection avoids this. Then, when the window is smaller, the features of I-frames cannot be used. We use the CU&PU features in the window as a comparison, which is used as a part of the features in [23], [24]. To further enhance the features, we also introduce the in-loop filtering (ILF) features used in [25] to form combined P-frame CU&PU&ILF features. We consider 2, 3, 5, and 8 consecutive frames as the small windows. The reason why we do not consider B-frames is that B-frames do not exist in all the profiles of encoding, and the SOTA methods did not utilize the features related to B-frames.

There are some details in the implementation. We only consider the nonoverlapping windows, otherwise more windows containing both non-transcoded and transcoded frames will be brought. When the number of remaining frames is less than the size of the window in the end, we regard the remaining frames as a window that is only used in testing. Similarly, windows containing both non-transcoded and transcoded frames are only used for testing. Since our method is trained and tested in a framewise fashion, the majority voting strategy is still used to obtain the detection result of a window. If the number of frames in the window is even and half of the frames are judged as positive samples, we treat the window as a positive sample. The performances of large and small windows are shown in Table XV and XIII. $FIoU$ is used as the criterion. SVM with the experimental optimal parameters is used as classifier for

TABLE XIV
FRAME-LEVEL PERFORMANCES OF PARTIAL TRANSCODING LOCALIZATION. 'SINGLE PU&DF' MEANS A ONE-PATH NETWORK WITH FUSED PU MAPS AND DECODED FRAMES AS THE INPUTS.

frame-level Acc	720P, HM		720P, x265		1080P, HM	
methods	4M	8M	4M	8M	4M	8M
Single PMP	0.8786	0.8725	0.8034	0.8146	0.8415	0.8020
Single DFP	0.7768	0.8051	0.8108	0.6738	0.6263	0.6047
Single PU&DF	0.8544	0.8566	0.7919	0.7893	0.8374	0.7703
Dual-path	**0.8881**	**0.8893**	**0.8112**	**0.8264**	**0.8599**	**0.8188**

the SOTA features. All the results are the average of three trainings and tests in a cross-content manner.

As one can observe from Table XV, [24], [25] and our method have their own strengths in different parameters. Although [24] performs slightly better, it is not pivotal because the results of large sizes are generally worse than those of small sizes by comparing Table XV and XIII. The large window results are even worse than the framewise results in some cases. This confirms that a large window is not the better choice. In Table XIII, the results of SOTA features increase with increasing size. This confirms that the effect of SOTA features is limited in small windows. However, when the small size increases to a certain level, the influence of the boundaries is obvious. For example, when the size increases to 8, the results of both SOTA features and our method decrease significantly. Unlike SOTA features, our method can achieve much better results in very small windows, which is advantageous because window with a size of 2 can minimize the influence of boundaries containing both non-transcoded and transcoded frames. It also proves that our framewise approach can make full use of limited information on a single frame. When the size is small (e.g., 2, 3, and 5), our method achieves generally fine results and performs better than SOTA features. Since the 1080P video contains more high-frequency details, especially when the bitrate is high, the transcoding artifacts will be affected. Although the performances of both SOTA and the proposed method decrease when processing 1080P videos, our performance still outperforms or rivals SOTA features in most cases.

6) Ablation study of the dual-path network: To clarify the role of two paths in the proposed network, the overall frame-level classification results in transcoding localization are shown in Table XIV. 'Single PU&DF' represents a one-path network with the fused PU map and decoded frame as the input. As one can observe, 'Single PU&DF' does not perform better because the PU map and decoded frame have no spatial correspondence, and the fused input will not provide extra artifacts for detection and localization. Therefore, the 'Single PU&DF' may not perform as well as a single path, as indicated by the underlined results in the Table XIV. However, our proposed dual-path network with separate inputs can fuse the functions of two paths instead of interfering with the overall performance. This is the difference between the one-path network with fused input and our dual-path network.

Similar to whole transcoding detection, the dual-path network performs slightly better than the single PMP. Although there is not much improvement, the reason is that DFP is used to distinguish AVC-HEVC from HEVC-HEVC in localization. The HEVC-HEVC frames also experience double compression, so the spatial artifacts are not as clear as naive HEVC. However, DFP is still indispensable once PMP is not effective, such as the case of using x265 with a low bitrate. Coupled with the adaptive fusion module, the dual-path network can make good use of both PMP and DFP.

7) Visualization: In addition to the quantitative comparisons, the visualization of framewise localization is also provided. Some examples are shown in Fig. 11. The abscissa represents frame index. Each 'stem' in the figures represents that the sample is judged as transcoded frame. Each figure incorporates the visualizations of three videos with the same content, namely, the transcoded part is located in the front, middle and rear of the video. The frames in the red boxes are the real transcoded frames. Please note that I-frames are not included in the figures. From Fig. 11(a) - 11(d), one can observe that relatively lower FA can present better visual effect in localization. The performances in IPPP 1080P videos and IPBB 720P videos are acceptable despite the drop of $FIoU$, as shown in Fig. 11(e) - 11(g). However, there are some unexpected detection cases with high FA and poor $recall$ that affects the localization, especially in IPBB 1080P videos, as shown in Fig. 11(h). Anyway, the evaluations have reported the advantages of our proposed method in dealing with partial transcoding detection and localization.

VI. DISCUSSION

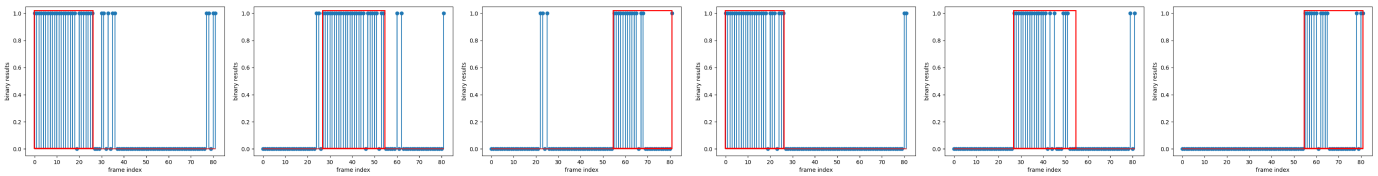
In this section, we will discuss whether the proposed method can be extended to other scenarios. According to the analysis in Sec. III-A, the mixing of HEVC clips with different coding standards (AVC and HEVC) will introduce the inconsistency of high-frequency components (or video quality). Therefore theoretically, our method can be extended to other HEVC localization scenarios where different coding parameters lead to inconsistent spatial quality of a mixed video. In video compression, in addition to the coding standards, the most important and direct parameter that can affect the video quality is the bitrate. If HEVC clips with low and high bitrates are merged and then the merged video is recompressed with a high bitrate, the clip with a fake bitrate is added with the irreversible distortion caused by the previous low bitrate. Therefore, our proposed method is expected to be useful in this scenario.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, to achieve not only whole transcoding detection but also partial transcoding localization, we propose a framewise scheme based on a dual-path CNN. A theoretical analysis is first conducted to explore the essential differences between non-transcoded and transcoded HEVC videos. Guided by the analysis in the spatial and compressed domains, we generate PU maps by introducing PU location information to make full use of the local PU information. Then, the decoded frames and PU maps are used as inputs of the network, which incorporates the specific convolutional modules to learn

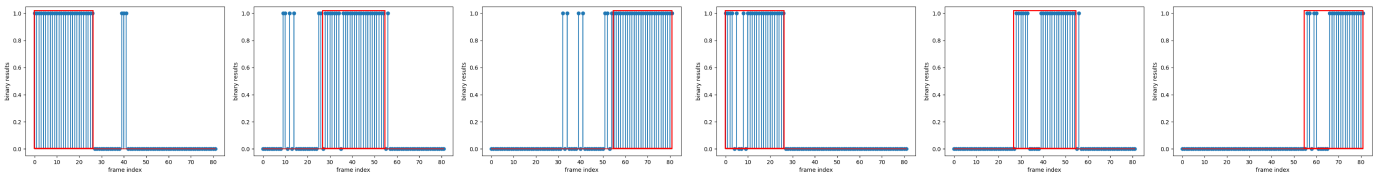
TABLE XV
THE $FIoU$ OF LARGE WINDOW BASED LOCALIZATION. 'WIN-SIZE' DENOTES THE SIZE OF WINDOWS. BOLD VALUES INDICATE BETTER RESULTS.

$FIoU$	720P, HM, 4M				720P, HM, 8M				720P, x265, 4M				720P, x265, 8M				1080P, HM, 4M				1080P, HM, 8M			
Win-Size	13	18	20	25	13	18	20	25	13	18	20	25	13	18	20	25	13	18	20	25	13	18	20	25
[23]	0.5199	0.5244	0.5509	0.4713	0.6076	0.5328	0.5744	0.4442	0.6082	0.5736	0.5773	0.6060	0.5367	0.5210	0.5724	0.5109	0.4823	0.4561	0.4970	0.4664	0.3323	0.2775	0.3024	0.3492
[24]	0.6077	0.5802	0.6036	**0.6110**	**0.6750**	0.6035	**0.6731**	0.5804	0.6339	**0.5855**	0.6182	**0.6096**	**0.6112**	0.5596	0.6122	0.4961	**0.5023**	0.4606	**0.5229**	0.4561	**0.4443**	0.4193	**0.4803**	**0.4080**
[25]	0.5468	0.4641	0.5467	0.5720	0.6348	0.5160	0.6474	0.5950	**0.6509**	0.4968	**0.6496**	0.6086	0.5489	0.5524	**0.6429**	0.5190	0.3290	0.2848	0.3905	0.3469	0.2965	0.3014	0.2826	0.2698
Proposed	**0.6139**	**0.5849**	**0.6149**	0.5602	0.6213	**0.6387**	0.6446	**0.6186**	0.5855	0.5659	0.5693	0.5415	0.5797	**0.5884**	0.5792	**0.5216**	0.4715	**0.4877**	0.4188	**0.4747**	0.4156	**0.4287**	0.3331	0.3548



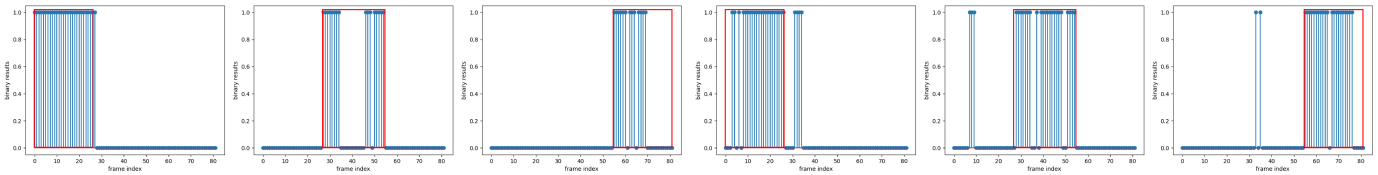
(a) Single PMP results: 720P, (JM+HM)-HM, IPPP, 8M *in_to_tree*

(b) Dual-path results: 720P, (JM+HM)-HM, IPPP, 8M *in_to_tree*



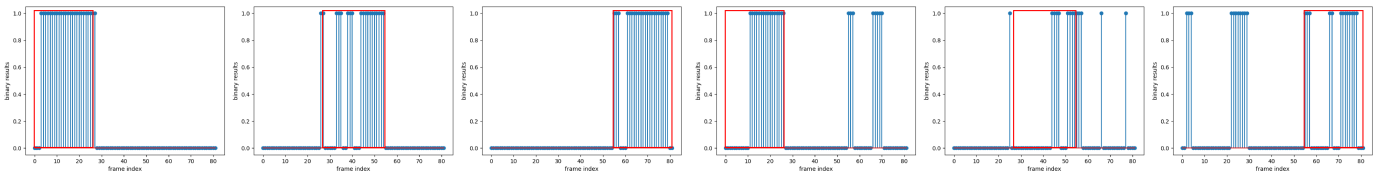
(c) Single PMP results: 720P, (JM+x265)-x265, IPPP, 8M *KristenAndSara*

(d) Dual-path results: 720P, (JM+x265)-x265, IPPP, 8M *KristenAndSara*



(e) 1080P, (x264+HM)-HM, IPPP, 4M *touchdown'* $FIoU(0.96, 0.57, 0.48)$

(f) 1080P, (x264+HM)-HM, IPPP, 8M *sunflower* $FIoU(0.70, 0.74, 0.72)$



(g) 720P, (JM+x265)-x265, IPBB, 8M *Fourpeople* $FIoU(0.85, 0.62, 0.81)$

(h) 1080P, (JM+x265)-x265, IPBB, 8M *speed_bag* $FIoU(0.45, 0.23, 0.34)$

Fig. 11. Some visual samples of transcoded frame localization through our proposed method. The abscissa represents frame index. 'Stem' represents that the sample is classified as transcoded frame. Each subfigure contains the visualizations of three videos with the same content, namely, the transcoded frames are located in the front, middle and rear. The frames in the red boxes are the real transcoded frames. (e)-(h) are the dual-path results.

feature representation. Finally, an adaptive fusion module is designed to obtain the optimal fused features of the two paths. Thus, our proposed network is capable of framewise detection and localization of the transcoded frames. Coupled with a voting strategy, the results of whole video transcoding detection can also be achieved. A large number of experiments have demonstrated the advantages of our proposed method compared with the existing methods, especially in the detection and localization of partial transcoding.

From the experimental results, the detection and localization on 720P videos are better than those on 1080P videos. Furthermore, it is more challenging for decoded frames in 1080P videos to be used in a deep-learning approach. In future work, more targeted theoretical analysis is needed to crop and identify the more distinguishable areas on spatial frames.

APPENDIX A
DIVISION OF TRAINING AND TESTING SETS

720P, Round1

Training: *mobcal*, *shields*, *stockholm*, *FourPeople*, *ducks_take_off*, *old_town_cross*, *vidyo4*. Test: *parkrun*, *Johnny*, *KristenAndSara*, *in_to_tree*, *park_joy*, *vidyo1*, *vidyo3*.

720P, Round2

Training: *shields*, *FourPeople*, *KristenAndSara*, *ducks_take_off*, *park_joy*, *old_town_cross*, *vidyo3*. Test: *mobcal*, *parkrun*, *stockholm*, *Johnny*, *in_to_tree*, *vidyo1*, *vidyo4*.

720P, Round3

Training: *parkrun*, *shields*, *stockholm*, *ducks_take_off*, *park_joy*, *vidyo3*, *vidyo4*. Test: *mobcal*, *FourPeople*, *Johnny*, *KristenAndSara*, *in_to_tree*, *old_town_cross*, *vidyo1*.

1080P, Round1

Training: *blue_sky*, *ducks_take_off*, *old_town_cross*, *pedestrian_area*, *rush_field_cuts*, *snow_mnt*, *station2*, *touchdown_pass*, *west_wind_easy*. Test: *crowd_run*, *in_to_tree*, *park_joy*, *riverbed*, *rush_hour*, *speed_bag*, *sunflower*, *tractor*.

1080P, Round2

Training: *crowd_run*, *ducks_take_off*, *park_joy*, *rush_hour*, *speed_bag*, *snow_mnt*, *sunflower*, *tractor*, *west_wind_easy*. Test: *blue_sky*, *in_to_tree*, *old_town_cross*, *pedestrian_area*, *riverbed*, *rush_field_cuts*, *station2*, *touchdown_pass*.

1080P, Round3

Training: *crowd_run*, *in_to_tree*, *park_joy*, *rush_hour*, *riverbed*, *pedestrian_area*, *station2*, *sunflower*, *tractor*. Test: *blue_sky*, *ducks_take_off*, *old_town_cross*, *rush_field_cuts*, *speed_bag*, *snow_mnt*, *touchdown_pass*, *west_wind_easy*.

ACKNOWLEDGMENT

The authors would like to thank the editors and reviewers for their careful work and valuable suggestions to this paper. The financial support of the Austrian Federal Ministry for Digital and Economic Affairs, the National Foundation for Research, Technology and Development, and the Christian Doppler Research Association is gratefully acknowledged. Christian Doppler Laboratory ATHENA: https://athena.itec.aau.at/.

REFERENCES

[1] P. Bestagini, K. M. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, and K. S. Tubaro, "An overview on video forensics," in *European signal processing conference*, 2012, pp. 1229–1233.

[2] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[3] R. D. Singh and N. Aggarwal, "Video content authentication techniques: a comprehensive survey," *Multimedia Systems*, vol. 24, no. 2, pp. 211–240, 2018.

[4] N. A. Shelke and S. S. Kasana, "A comprehensive survey on passive techniques for digital video forgery detection," *Multimedia Tools and Applications*, vol. 80, no. 4, pp. 6247–6310, 2021.

[5] Z. Huang, F. Huang, and J. Huang, "Detection of double compression with the same bit rate in MPEG-2 videos," in *2014 IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*. IEEE, 2014, pp. 306–309.

[6] Z. Zhang, J. Hou, Y. Zhang, J. Ye, and Y. Shi, "Detecting multiple H.264/AVC compressions with the same quantisation parameters," *IET Information Security*, vol. 11, no. 3, pp. 152–158, 2016.

[7] X. Jiang, P. He, T. Sun, F. Xie, and S. Wang, "Detection of double compression with the same coding parameters based on quality degradation mechanism analysis," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 1, pp. 170–185, 2018.

[8] 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, 2020.

[9] 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*. ACM, 2006, pp. 37–47.

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

[11] 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.

[12] 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. 23, pp. 3179–3192, 2020.

[13] H. Yao, R. Ni, and Y. Zhao, "Double compression detection for H.264 videos with adaptive GOP structure," *Multimedia Tools and Applications*, vol. 79, no. 9, pp. 5789–5806, 2020.

[14] Y. Su and J. Xu, "Detection of double-compression in MPEG-2 videos," in *2nd International Workshop on Intelligent Systems and Applications*. IEEE, 2010, pp. 1–4.

[15] 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.

[16] Y. Yu, H. Yao, R. Ni, and Y. Zhao, "Detection of fake high definition for HEVC videos based on prediction mode feature," *Signal Processing*, vol. 166, p. 107269, 2020.

[17] 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, 2019.

[18] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards including high efficiency video coding HEVC," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1669–1684, 2012.

[19] H. Yuan, C. Guo, J. Liu, X. Wang, and S. Kwong, "Motion-homogeneous-based fast transcoding method from H.264/AVC to HEVC," *IEEE Transactions on Multimedia*, vol. 19, no. 7, pp. 1416–1430, 2017.

[20] J. Xu, M. Xu, Y. Wei, Z. Wang, and Z. Guan, "Fast H.264 to HEVC transcoding: A deep learning method," *IEEE Transactions on Multimedia*, vol. 21, no. 7, pp. 1633–1645, 2018.

[21] A. Costanzo and M. Barni, "Detection of double AVC/HEVC encoding," in *24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016, pp. 2245–2249.

[22] L. Yu, Z. Zhang, X. Yang, and Z. Li, "P frame PU partitioning mode based H.264 to hevc video transcoding detection," *Journal of Applied Sciences*, vol. 36, pp. 278–286, 2018.

[23] S. Bian, H. Li, T. Gu, and A. Kot, "Exposing video compression history by detecting transcoded hevc videos from AVC coding," *Symmetry*, vol. 11, no. 67, 2019.

[24] Z. Zhang, C. Liu, Z. Li, L. Yu, and H. Yan, "Detection of transcoding from H.264/AVC to HEVC based on CU and PU partition types," *Symmetry*, vol. 11, no. 1343, 2019.

[25] Q. Xu, X. Jiang, T. Sun, and A. C. Kot, "Detection of transcoded HEVC videos based on in-loop filtering and PU partitioning analyses," *Signal Processing: Image Communication*, vol. 92, p. 116109, 2021.

[26] Z. Tang, L. Chen, H. Yao, X. Zhang, and C. Yu, "Video hashing with DCT and NMF," *The Computer Journal*, vol. 63, no. 7, pp. 1017–1030, 2020.

[27] Z. Tang, S. Zhang, X. Zhang, Z. Li, Z. Chen, and C. Yu, "Video hashing with secondary frames and invariant moments," *Journal of Visual Communication and Image Representation*, vol. 79, p. 103209, 2021.

[28] W. Pu, J. Chen, K. Rapaka, X. Li, and M. Karczewicz, "High frequency SAO for scalable extension of HEVC," in *2013 Picture Coding Symposium (PCS)*. IEEE, 2013, pp. 121–124.

[29] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Transactions on systems, man, and cybernetics*, no. 6, pp. 610–621, 1973.

[30] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[32] M. Boroumand, M. Chen, and J. Fridrich, "Deep residual network for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1181–1193, 2018.

[33] M. Chen, V. Sedighi, M. Boroumand, and J. Fridrich, "Jpeg-phase-aware convolutional neural network for steganalysis of JPEG images," in *Proceedings of the 5th ACM Workshop on Information Hiding and Multimedia Security*, 2017, pp. 75–84.

[34] J. Ye, J. Ni, and Y. Yi, "Deep learning hierarchical representations for image steganalysis," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 11, pp. 2545–2557, 2017.

[35] Xiph.org Video Test Media[derf's collection]. Available online: https://media.xiph.org/video/derf/.

[36] JM Software. Available online: http://iphome.hhi.de/suehring/tml/.
[37] HM Software. Available online: https://hevc.hhi.fraunhofer.de/trac/hevc/browser/tags.
[38] X265 Software. Available online: https://github.com/videolan/x265.
[39] FFmpeg. Available online: http://ffmpeg.org/.
[40] PyTorch. Available online: https://pytorch.org.
[41] X264 Software. Available online: https://www.videolan.org/developers/x264.html.
[42] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

Haichao Yao received the B.S. degree from Changchun University of Science and Technology, China, in 2016. He is currently pursuing the Ph.D. degree in Institute of Information Science, Beijing Jiaotong University, Beijing, China. His research interests include multimedia forensics and video compression.

Rongrong Ni received the B.S. degree and the Ph.D. degree from Beijing Jiaotong University (BJTU) in China, in 1998 and 2005, respectively. Since 2005, she has been the faculty of the School of Computer and Information Technology, and the Institute of Information Science, BJTU, where she is a Professor since 2013. Her current research interests include image processing, multimedia forensics and digital watermarking.

Hadi Amirpour is a postdoctoral researcher at ATHENA (https://athena.itec.aau.at) directed by Prof. Christian Timmerer. He received his B.Sc. degrees in Electrical and Biomedical Engineering, and he pursued his M.Sc. in Electrical Engineering. He got his Ph.D. in computer science from the University of Klagenfurt in 2022. He was appointed co-chair of Task Force 7 (TF7) Immersive Media Experience (IMEx) at the 15th Qualinet meeting. He was involved in the project EmergIMG, a Portuguese consortium on emerging imaging technologies, funded by the Portuguese funding agency and H2020. Currently, he is working on the ATHENA project in cooperation with its industry partner Bitmovin. His research interests are image processing and compression, video processing and compression, quality of experience, emerging 3D imaging technology, and medical image analysis. Further information at hadiamirpour.github.io.

Christian Timmerer (M'08-SM'16) is an associate professor at the Institute of Information Technology (ITEC) and is the director of the Christian Doppler (CD) Laboratory ATHENA (https://athena.itec.aau.at). His research interests include immersive multimedia communication, streaming, adaptation, and quality of experience where he co-authored seven patents and more than 200 articles. He was the general chair of WIAMIS 2008, QoMEX 2013, MMSys 2016, and PV 2018 and has participated in several EC-funded projects, notably DANAE, ENTHRONE, P2P-Next, ALICANTE, SocialSensor, COST IC1003 QUALINET, and ICoSOLE. He also participated in ISO/MPEG work for several years, notably in the area of MPEG-21, MPEG-M, MPEG-V, and MPEG-DASH where he also served as standard editor. In 2013 he cofounded Bitmovin (http://www.bitmovin.com/) to provide professional services around MPEG-DASH where he holds the position of the Chief Innovation Officer(CIO)—Head of Research and Standardization. Further information at http://timmerer.com.

Yao Zhao received the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1996. He became an Associate Professor with BJTU in 1998 and became a Professor in 2001. From 2001 to 2002, he was a Senior Research Fellow with the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. In 2015, he visited the Swiss Federal Institute of Technology, Lausanne, Switzerland (EPFL). From 2017 to 2018, he visited the University of Southern California. He is currently the Director of the Institute of Information Science, BJTU. His current research interests include image/video coding, digital watermarking and forensics, video analysis and understanding, and artificial intelligence. Dr. Zhao serves or served on the Editorial Boards of several international journals, including as an Associate Editor of the IEEE TRANSACTIONS ON CYBERNETICS, an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, a Senior Associate Editor of the IEEE SIGNAL PROCESSING LETTERS, and an Area Editor of the Signal Processing: Image Communication. He was named as a Distinguished Young Scholar by the National Science Foundation of China in 2010, and was elected as a Chang Jiang Scholar of the Ministry of Education of China in 2013.