

RESEARCH

Open Access



# HEVC double compression detection under different bitrates based on TU partition type

Lifang Yu<sup>1</sup>, Yiyuan Yang<sup>2</sup>, Zhaohong Li<sup>3\*</sup> , Zhenzhen Zhang<sup>1</sup> and Gang Cao<sup>4</sup>

## Abstract

During the process of authenticating the integrity of digital videos, double compression is an important evidence. High-efficiency video coding (HEVC) is the latest coding standard proposed in 2013 and has shown superior performance over its predecessors. It brings in several novel syntactic units, such as coding tree unit (CTU), prediction unit (PU), and transform unit (TU). Few methods have been reported to detect HEVC double compression by utilizing such new characteristics. In this paper, a novel scheme based on TU is proposed to detect double compression of HEVC videos. The histogram of each TU partition type in the first I/P frame of all GOPs is calculated. The feature set is fed into SVM to classify single and double compressed videos. Experimental results have demonstrated the effectiveness of TU-based feature (accuracy in [0.84, 0.97]) that bears very low dimension (10-D). And it is also shown that TU-based feature set can be combined with other feature sets to boost their performance.

**Keywords:** HEVC, Double compression detection, TU partition type, Histogram, Bitrate

## 1 Introduction

Nowadays, video editing software can be easily accessed. At the same time, video editing algorithms have been proposed by researchers [1]. The editing of videos becomes easier and easier. On the one hand, video editing helps people to achieve astounding visual effect in some situations, e.g., movies and video advertisements. On the other hand, editing videos with malicious purpose may lead to potentially serious moral, ethical, and legal consequences if tampered videos are considered as evidences. Hence, it is important to verify the integrity and authenticity of videos.

Double compression is a distinction of modified videos. Video tampering is commonly fulfilled in non-compressed domain. The original video will first be decompressed to frames, and the manipulation is performed directly on frames. Then, the tampered frames are recompressed [2].

High-efficiency video coding (HEVC) is the latest generation of video coding standard prepared by the Joint collaborative Team on Video Coding [3]. Although research works on double compression detection are prosperous on video standards preceding HEVC [4–10], e.g., MPEG,

H.264, the research of double compression detection on HEVC needs more attention. Huang et al. [11] utilize co-occurrence matrixes of discrete cosine transform (DCT) coefficients to detect double compressed HEVC videos when the quantization parameters (QP) change in the process of the second compression. In 2016, our group proposed a feature set to detect double compression of HEVC with the same QP after double compression [12]. The prediction unit (PU) partitioning type, which is a specific syntactic unit owned by HEVC, was firstly utilized and showed a promising detection capability. Xu et al. [13] use the sequence of number of prediction unit of its prediction mode (SN-PUPM) to conduct double compression detection of HEVC videos in the situation that the GOP size changes after double compression.

When considering transferring through the network, adopting bitrate as the ultimate compression parameter is more suitable because of the limit of band width. In this situation, videos are compressed at a certain bitrate. After they are tampered, they will be recompressed at another bitrate, which can be equal or not equal to the bitrate used in the first compression. In the process of video encoding, QP is assigned by the rate-distortion control process and it may change after double compression. There has been some research works engaged in HEVC double compression detection with designated

\* Correspondence: [zhhliz@bjtu.edu.cn](mailto:zhhliz@bjtu.edu.cn)

<sup>3</sup>School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China

Full list of author information is available at the end of the article

bitrate. Li et al. [14] proposed a 164-D (dimension) feature set based on co-occurrence matrix of PU types and DCT coefficients of I frame. Liang et al. [15] proposed a 25-D feature set based on the histogram of PU partition types of P frame.

In this work, we focus on the situation with designated bitrate and detecting HEVC double compression with different bitrates. We investigate transform unit (TU) partition type of single and double compressed videos and employ the histogram of the TU partition type in the first I and P frame of every group of pictures (GOP). As stated in Section 5, our proposed TU-based feature is effective in detecting double HEVC compression. Moreover, when combined with our TU-based feature set, the detection performance of some other feature sets will be boosted as shown in experiments.

The rest of this paper is organized as follows. In Section 2, we introduce backgrounds simply. Then, in Section 3, the feature set adopted in this paper is stated in details. Experimental results together with a comparison with a former work are shown in Section 4 to verify the effectiveness of our feature set. Finally, discussion and conclusions are made in Section 5.

## 2 Related work

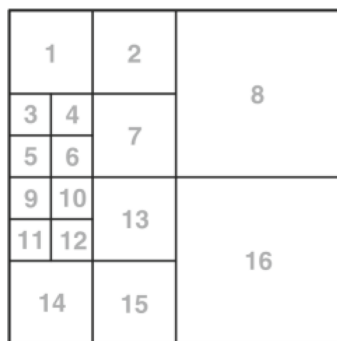
### 2.1 Background of transforming unit (TU)

Compared to other members in the family of video coding standards, HEVC has the highest feasibility. More than half of the average bitrate savings of HEVC relative to its predecessor H.264/MPEG-4 AVC can be attributed to its increased flexibility of block partitioning for prediction and transform coding [16].

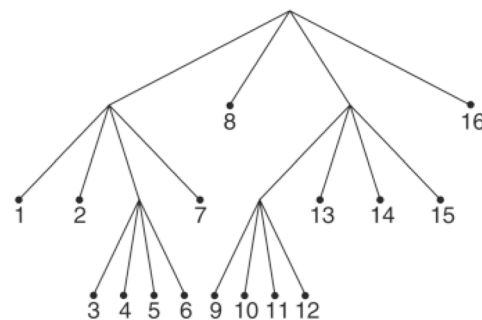
In recent years, the industry of standard- and high-definition (HD) video has been developed vigorously. One feature of large-sized images is that the area of smooth regions is larger. An encoding with larger blocks can greatly improve the encoding efficiency. Considering the characteristics of HD videos, the coding tree unit (CTU) is introduced in H.265/HEVC. For a luma CTU with  $L \times L$  luma samples,  $L$  can be chosen from the values 16, 32, and 64. As shown in Fig. 1a, an image can be divided into non-overlapping CTUs. Each CTU will be further divided into coding units (CU) using quadtree (as shown in Fig. 1b, c. A CTU can be also directly used as a CU. Therefore, the size of the CU is a variable, and the largest one is  $64 \times 64$  luma samples, while the smallest one is  $8 \times 8$  luma samples. On the one hand, a CU

CTU <sub>0</sub>	CTU <sub>1</sub>	CTU <sub>2</sub>	CTU <sub>3</sub>	CTU <sub>4</sub>	CTU <sub>5</sub>	CTU <sub>6</sub>
CTU <sub>7</sub>	CTU <sub>8</sub>	CTU <sub>9</sub>	CTU <sub>10</sub>	CTU <sub>11</sub>	CTU <sub>12</sub>	CTU <sub>13</sub>
CTU <sub>14</sub>	CTU <sub>15</sub>	CTU <sub>16</sub>	CTU <sub>17</sub>	CTU <sub>18</sub>	CTU <sub>19</sub>	CTU <sub>20</sub>
CTU <sub>21</sub>	CTU <sub>22</sub>	CTU <sub>23</sub>	CTU <sub>24</sub>	CTU <sub>25</sub>	CTU <sub>26</sub>	CTU <sub>27</sub>

**a**

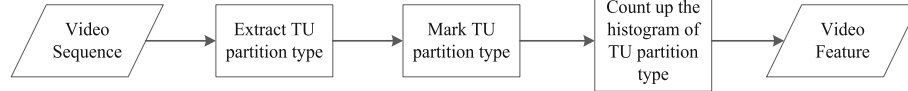


**b**



**c**

**Fig. 1** Examples of the partitioning of a picture into CTUs (a) and the partitioning of a  $64 \times 64$  coding tree unit (CTU) into coding units (CUs) (b). The partitioning can be illustrated by a quadtree (c), where the numbers indicates the coding order of CUs



**Fig. 2** The procedure of feature extraction

of a large size enables great increasing in the coding efficiency of a smooth region. On the other hand, small CUs are good at dealing with local image details, which makes the prediction of complex areas more accurate.

Transform unit (TU) is the basic unit of transformation and quantization operation, and its size is also flexible. A CU is recursively divided into TUs based on a quadtree approach. The minimum size of a TU is  $4 \times 4$ , and the maximum size is  $64 \times 64$ . TUs with larger sizes concentrate energy better. And TUs with smaller sizes save more image details. This flexible partitioning structure makes the residual energy be fully compressed in the transform domain, which further improves the coding gain. In fact, a TU with the size of  $64 \times 64$  implies that it must be further divided into four  $32 \times 32$  TUs, because the largest size of DCT (discrete cosine transform) transform operation is  $32 \times 32$ . In this paper, the sizes of TUs are regarded in five types, including  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$ ,  $32 \times 32$ , and  $64 \times 64$ , regardless of implicit further division.

### 3 Methods—TU partition type features

In this section, we will introduce our proposed scheme in details. First, we introduce the feature set based on TU partition type. Then, we analyze the effectiveness of the proposed feature set.

#### 3.1 Feature extraction method

In this subsection, we examine the unique coding structure of HEVC. More specifically, the TU partition type of the first I/P frame in each GOP is investigated, and the histogram of TU partition type in the first I/P frame in all GOPs is used as the feature of video classification. We only take luminance component of TUs into account during feature extraction process.

The process of feature extraction can be divided into three steps, and its flow chart is shown in Fig. 2. Firstly, we extract the TU partition type of video frames using video analysis software Gitl\_HEVC\_Analyzer [17]. Secondly, we mark the TU partition type of video frames. Based on blocks with sizes of  $8 \times 8$ , the TU partition

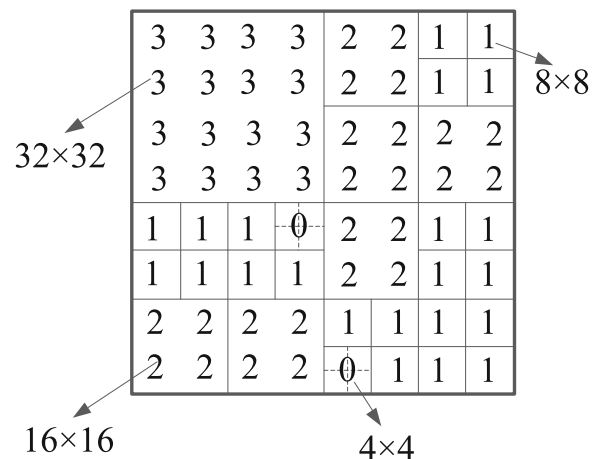
types are marked. Table 1 shows the label for the TU partition types. Figure 3 shows an example of marking TU partition types with their labels in a  $64 \times 64$  CU. Thirdly, we count up the histogram of TU partition type. The number of each TU partition type in the first I/P frame of each GOP is counted, denoted by  $F_i = \{f_{i,0}, f_{i,1}, f_{i,2}, f_{i,3}, f_{i,4}\} (i = 1, \dots, M)$ , where  $M$  is the number of GOPs in the video sequence, and the value of the second index of  $f$  is the same as the label shown in Table 1. Each  $F_i$  records the number of  $8 \times 8$  blocks corresponding to the five TU partition types in the  $i$ -th GOP. The average value of  $F_i$  is calculated as:

$$F = \frac{\sum_{i=1}^M F_i}{M} \quad (1)$$

$F$  is a 5-D vector, which can be denoted as  $F = \{f_0, f_1, f_2, f_3, f_4\}$ . Then, each element of is normalized as:

$$h_k = \frac{f_k}{\sum_{k=0}^4 f_k} \quad (2)$$

where  $k = \{0, \dots, 4\}$ . The histogram of TU partition type can be denoted as  $H = \{h_0, h_1, h_2, h_3, h_4\}$ .



**Fig. 3** Example of marking TU partition types with their labels in a  $64 \times 64$  CU

**Table 1** Labels for the TU partition types

Label	0	1	2	3	4
Type	$4 \times 4$	$8 \times 8$	$16 \times 16$	$32 \times 32$	$64 \times 64$

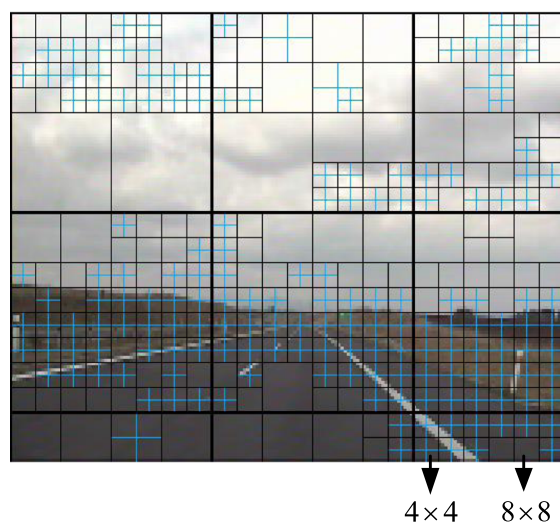
The histogram of TU partition type of the first I frame in all GOPs is called *TU-I* in the rest of this paper, and that of the first P frame is called *TU-P*. TU-I is a 5-D vector, so does TU-P. Then, in total, we have a 10-D vector corresponding to TU partition type, which will be referred as *TU-IP* in the following text. The TU-IP feature describes the holistic distribution of the TU partition type in the first I/P frame of all GOPs, and it is the unique characteristic of HEVC.

### 3.2 Effectiveness of TU-based feature

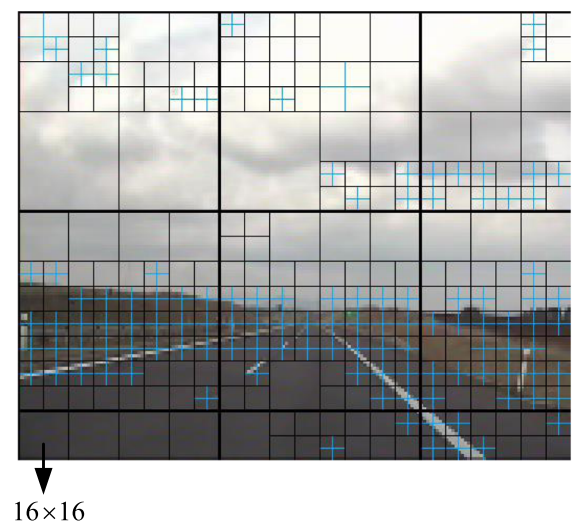
In this subsection, we will analyze the effectiveness of TU-based features. Figure 4 shows the TU partition type in the first I frame and the first P frame of a GOP in a single and

double compressed. The black solid box represents the CU partitioning type. If there are blue lines inside the black solid box, it means the CU is partitioned into several TUs recursively using quadtree. Or else, TU bears the same size as CU. From Fig. 4, we can see that there is an obvious difference in the TU partition type in the first I and P frames of a GOP between a single compressed video and its corresponding double one.

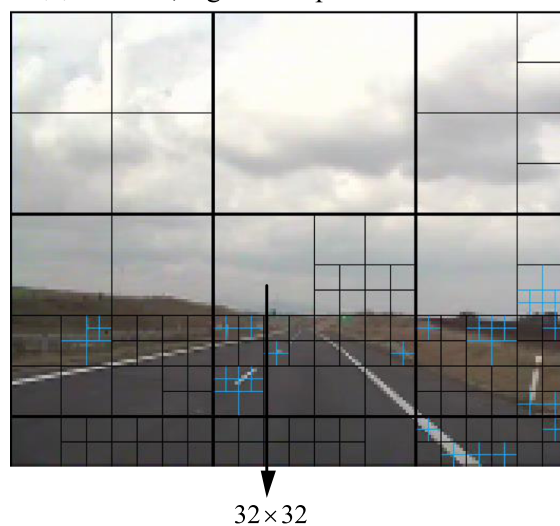
Figure 5 shows the average number of each TU partition type in the first I/P frame of all GOPs of a single compressed video and its corresponding double compressed one. The average number of each TU partition type is the parameter  $F$  in Eq. (1). From the figure, we can observe obvious difference between the TU partition type of the single



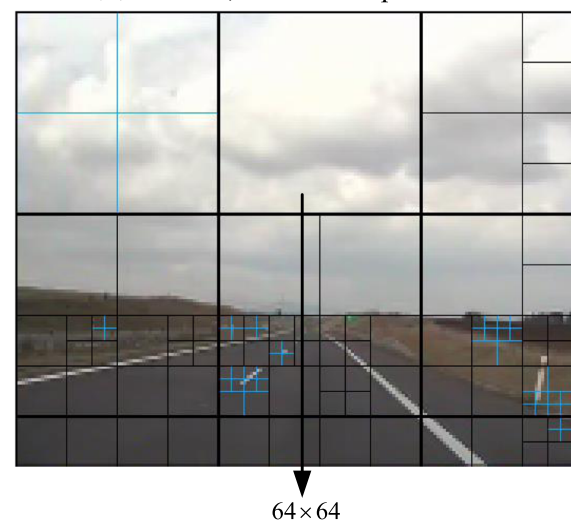
(a) I frame, signal compressed



(b) I frame, double compressed

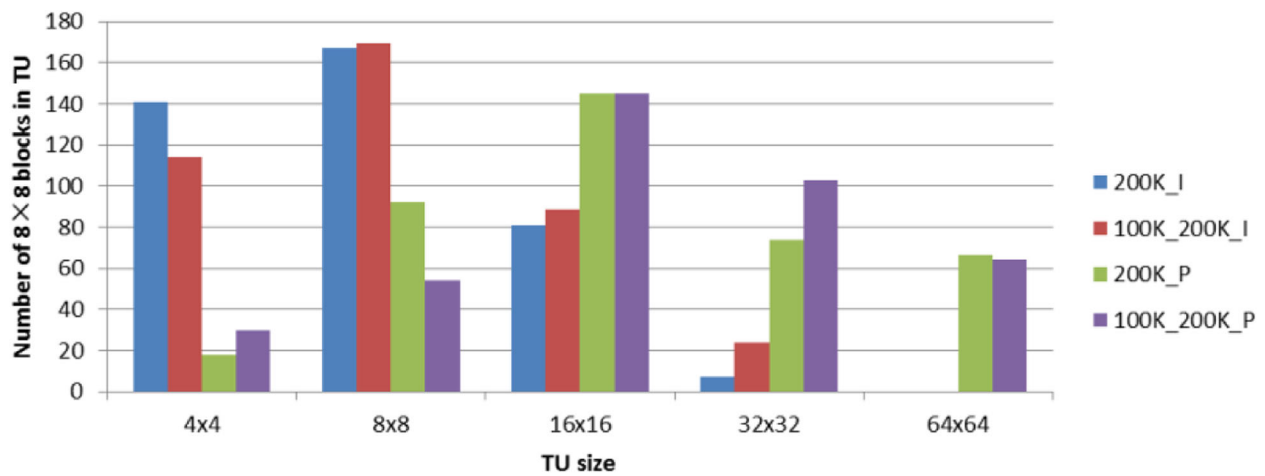


(c) P frame, signal compressed



(d) P frame, double compressed

**Fig. 4 a–d** Example of TU partition type in the first I frame and the first P frame of a GOP in a single compressed video and its corresponding double compressed video



**Fig. 5** The average number of each TU partition type in the first I/P frame of all GOPs of single (200 k) and double (100 k–200 k) compressed videos. '200k\_I' ('200k\_P') means I (P) frame of a single compressed video with bitrate 200 k. '100\_200k\_I' ('100\_200k\_P') means I (P) frame of a double compressed video with bitrate 100 k in the first compression process and 200 k in the second compression process. The video for testing is the first part of video named "akiyo" in QCIF database. Information on video database can be found in Section 4

and double compressed videos. For example, there is a big difference between the blue and red bar when TU size is  $4 \times 4$ , and this is also true for the green and purple bar when TU size is  $8 \times 8$ .

## 4 Results and discussion

### 4.1 Experimental setup

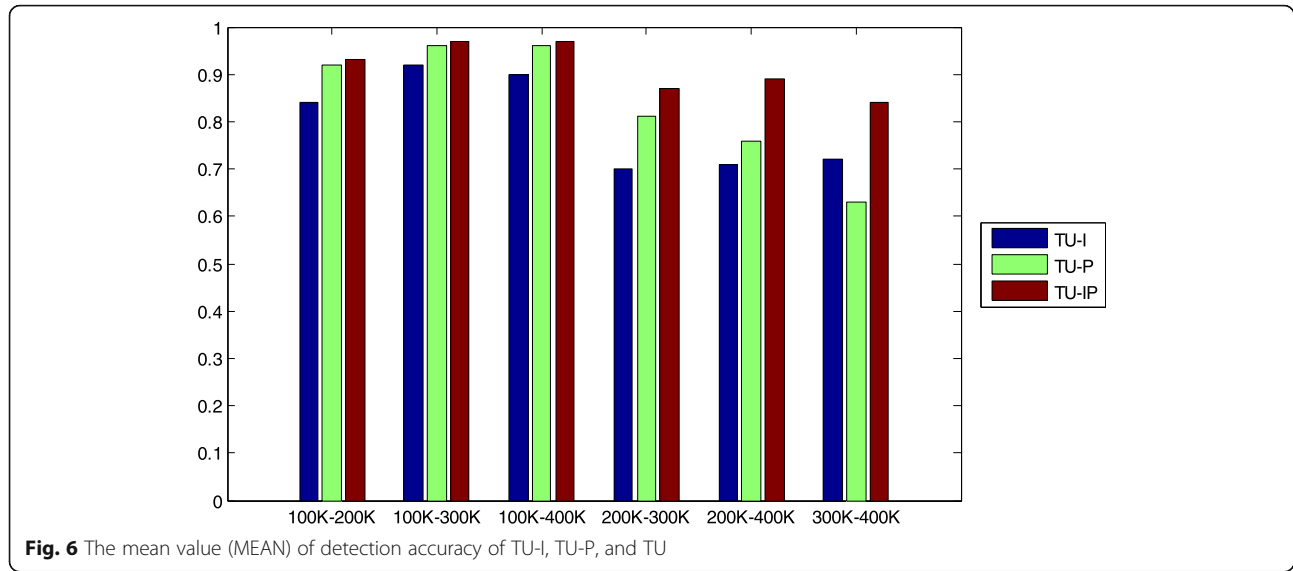
In experiments, we use 17 uncompressed YUV sequences in QCIF format [18] as the initial video, whose resolution is  $176 \times 144$ . We also use 18 CIF format [19] videos with the resolution of  $352 \times 288$ . To increase the size of the video

database, each video is divided into non-overlapping fragments with lengths of 100 frames. And a total of 36 QCIF video fragments and 43 CIF video fragments are obtained.

To obtain single compressed video database, we compress the video fragments to HEVC format at bitrate  $B_2$ . And to obtain double compressed video database, we compress the video fragments to HEVC format at bitrate  $B_1$ , then after decompressing we recompress them at bitrate  $B_2$ . The bitrate group ( $B_1 - B_2$ ) is set to be the following values:  $\{(100 - 200), (100 - 300), (100 - 400), (200 - 300), (200 - 400)\}$  Kbps. In all these encoding and decoding

**Table 2** Detection accuracy of the double compressed video fragments from its corresponding single compressed ones using TU-I, TU-P, and TU features on QCIF video dataset

$B_1 - B_2$ (Kbps)	100–200	100–300	100–400	200–300	200–400	300–400
TU-I(5D)						
MEAN	0.84	0.92	0.90	0.70	0.71	0.72
STDEV	0.10	0.07	0.07	0.10	0.09	0.09
MED	0.83	0.92	0.92	0.67	0.71	0.71
MAD	0.08	0.08	0.08	0.08	0.04	0.04
TU-P(5D)						
MEAN	0.92	0.96	0.96	0.81	0.76	0.63
STDEV	0.06	0.05	0.06	0.10	0.09	0.11
MED	0.92	1.00	1.00	0.83	0.75	0.67
MAD	0.00	0.00	0.00	0.08	0.08	0.08
TU-IP(10D)						
MEAN	0.93	0.97	0.97	0.87	0.89	0.84
STDEV	0.07	0.06	0.06	0.1	0.08	0.11
MED	0.92	1.00	1.00	1.00	0.92	0.83
MAD	0.08	0.00	0.00	0.10	0.05	0.08



process, we utilize the codec HM10.0 [20]. In the process of encoding, frame rate, intra period, and GOP size are the three main parameters, and they are set to be 30, 4, and 4, respectively. Thus, the total number of first P frame in a video fragment is 25, and that of the first I frame is also 25.

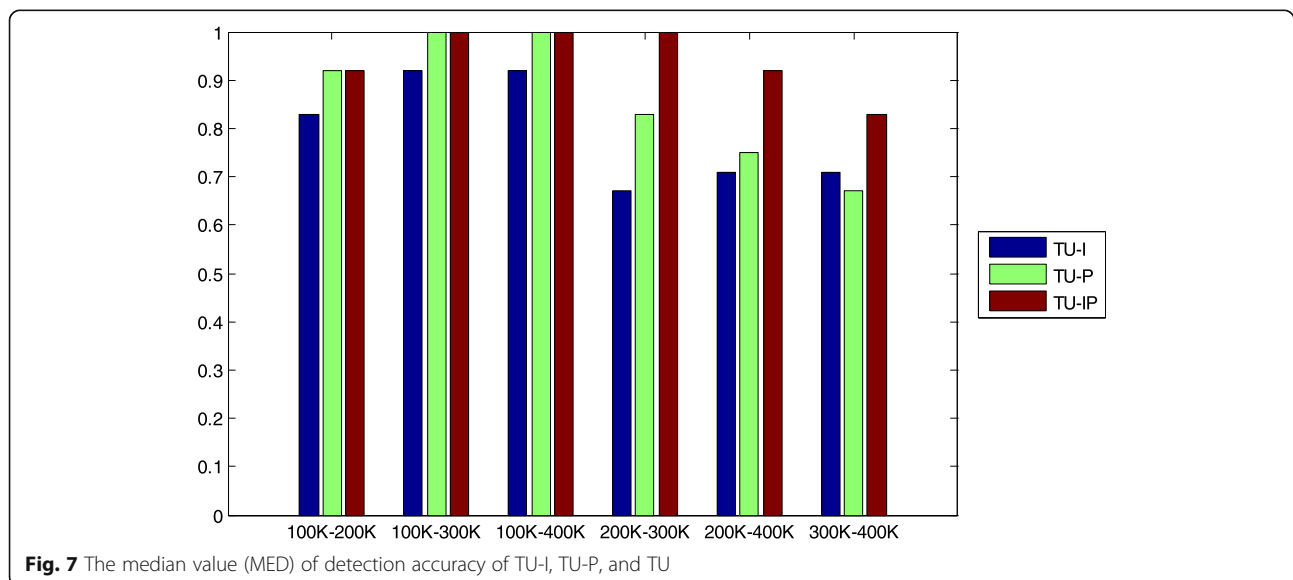
The detection results are expressed in terms of the detection probability  $P_{\text{accuracy}}$ . When the cardinalities of the positive set and the negative set are the same, which is the scenario in our experiments,  $P_{\text{accuracy}}$  is defined as follows:

$$P_{\text{accuracy}} = \frac{P_{\text{TP}} + P_{\text{TN}}}{2} \quad (3)$$

where  $P_{\text{TP}}$  and  $P_{\text{TN}}$  are the rate of true positive and true negative, respectively.

All classifiers presented in this paper were constructed by using libSVM [21] with the polynomial kernel  $k(x, y) = (\gamma x^T y + \text{coef0})^d$ ,  $\gamma > 0$ .  $\text{coef0}$  and  $d$  are set to be the default value 0 and 3, respectively. The classifier of DCT136 [10] is the only exception, where the classifier is libSVM with the rbf kernel as in the original paper.

Before training the libSVM on the training set, the value of the penalization parameter  $C$  and the kernel parameter need to be set. These hyper-parameters balance the complexity and accuracy of the classifier. The hyper-parameter  $C$  penalizes the error on the training set. Higher values of  $C$  produce classifiers more accurate on the training set but also more complex with a possibly worse generalization. On the other hand, a smaller value of  $C$  produces simpler classifiers with worse accuracy on the



**Table 3** The parameters used for training libSVM for feature set TU-I, TU-P, and TU-IP on QCIF video dataset

$B_1 - B_2$ (Kbps)	100–200	100–300	100–400	200–300	200–400	300–400
TU-I(5D)						
c	2.8284	5.6569	1.0000	1.0000	16.0000	5.6569
g	2.8284	11.3137	16.0000	4.0000	0.0884	1.4142
TU-P(5D)						
c	4.0000	11.3137	2.8284	1.4142	1.0000	16.0000
g	8.0000	0.3536	0.5000	8.0000	5.6569	16.0000
TU-IP(10D)						
c	5.6569	8.0000	2.0000	2.0000	2.8284	0.7071
g	1.0000	0.7071	1.0000	16.0000	4.0000	11.3137

training set but hopefully with better generalization. The role of the kernel parameter is similar to  $C$ : Higher values of  $\gamma$  make the classifier more pliable but likely prone to over-fitting the data, while lower values of  $\gamma$  have the opposite effect.

The values of  $C$  and  $\gamma$  should be chosen to give the classifier the ability to generalize. The standard approach is to estimate the error on unknown samples using cross-validation on the training set on a fixed grid of values and then select the value corresponding to the lowest error. In this paper, we used five-fold cross-validation with the multiplicative grid

$$C \in \{2^i | i \in \{-2, -1.5, -1, \dots, 3, 3.5, 4\}\}$$

$$\gamma \in \{2^i | i \in \{-4, -3.5, -3, \dots, 3, 3.5, 4\}\}.$$

$C$  and  $\gamma$  pair that can get the highest accuracy with the smallest  $C$  is chosen and denoted as  $(C_0, \gamma_0)$ .

We randomly split video set into training and testing sets. Then,  $(C_0, \gamma_0)$  is used to train the model on the training set, and the predicted label of testing set is obtained based on the trained model. This process is

executed by 100 times, and statistics of the detection accuracy will be shown in the tables or/and figures. We use four kinds of statistics, that is, mean value (MEAN), standard deviation (STDEV), median value (MED), and median absolute deviation (MAD). For QCIF dataset, 30 single compressed video fragments and their corresponding double compressed video fragments are used for training, and the rest 6 single compressed video fragments and their corresponding double compressed video fragments are used for testing. For CIF dataset, the size of training and testing sets are 36 and 7, respectively.

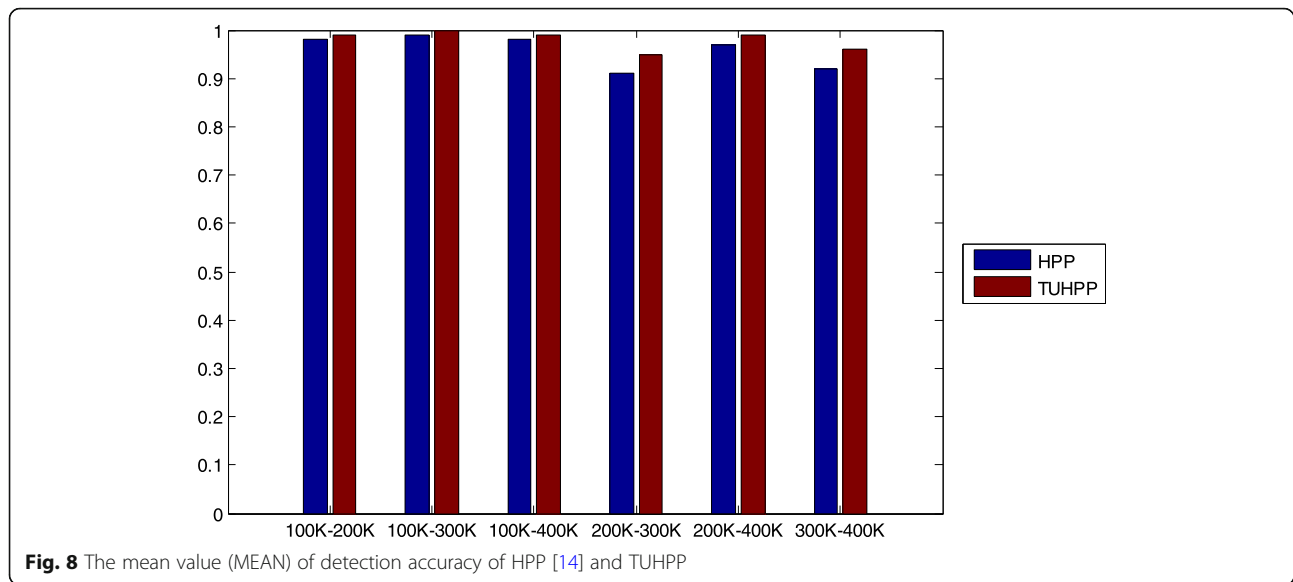
#### 4.2 Evaluating the effectiveness of TU features

TU-IP features are formed from TU partition type of the first I/P frame in all GOPs. It is the specific feature of HEVC. In this subsection, we will show the effectiveness of the TU-IP features in detecting double compression of HEVC.

Table 2 shows the detection accuracy of the double compressed video fragments from its corresponding single compressed ones. It can be seen that TU-I or TU-P

**Table 4** Detection accuracy of the double compressed video fragments from its corresponding single compressed ones using HPP feature and TUHPP feature on QCIF video dataset

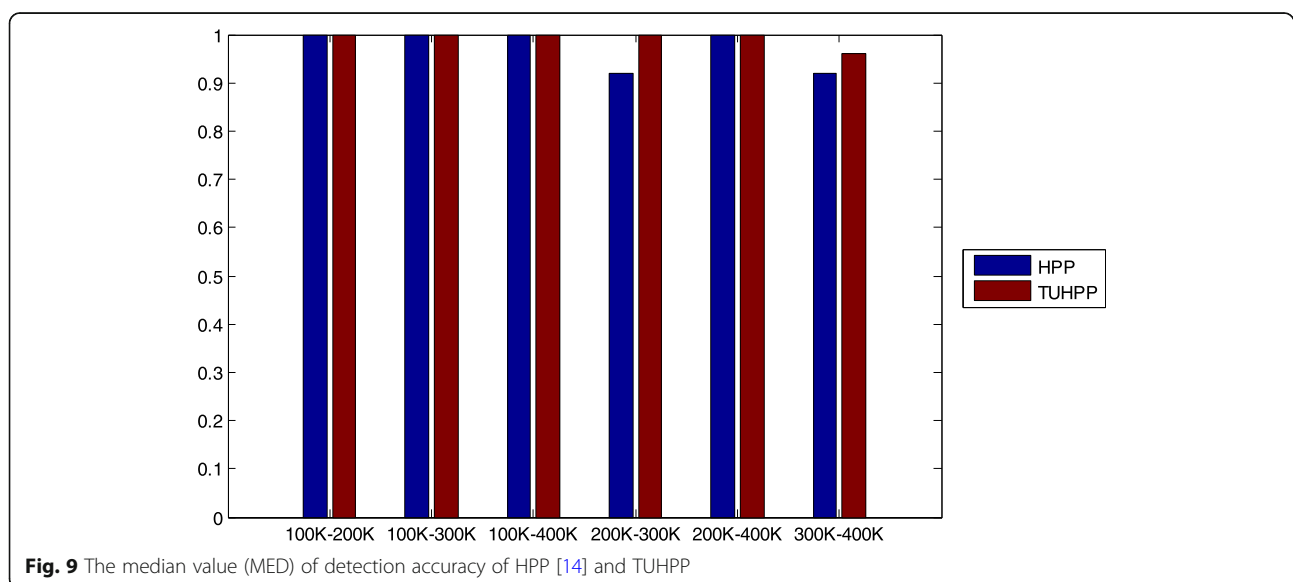
$B_1 - B_2$ (Kbps)	100–200	100–300	100–400	200–300	200–400	300–400
HPP(25D)						
MEAN	0.98	0.99	0.98	0.91	0.97	0.92
STDEV	0.04	0.03	0.05	0.08	0.06	0.07
MED	1.00	1.00	1.00	0.92	1.00	0.92
MAD	0.00	0.00	0.00	0.08	0.00	0.08
TUHPP(35D)						
MEAN	0.99	1.00	0.99	0.95	0.99	0.96
STDEV	0.03	0.00	0.02	0.06	0.03	0.05
MED	1.00	1.00	1.00	1.00	1.00	0.96
MAD	0.01	0.00	0.01	0.05	0.01	0.04



individually shows effectiveness in detecting HEVC double compression. For TU-I, the MEAN ranges from 0.70 to 0.92, and for TU-P, it ranges from 0.63 to 0.96. The MEAN of TU lies in [0.74, 0.97]. MED of TU-I and TU-P lies in [0.67, 0.92] and [0.67, 1], respectively, and that of TU lies in [0.75, 1]. From Figs. 6 and 7, we can observe easily that TU outperforms TU-I and TU-P in both the mean value and the median value of the detection accuracy of HEVC double compression. Merging TU-I and TU-P together shows conspicuous superiority over any of them. When evaluating the detection capability of TU-I and TU-P, we should note that the dimension of TU-I or TU-P is only 5-D and that of TU is only 10-D. The dimension of features based on TU is

low compared to other features for HEVC double compression detection, e.g., DCT136 [10] and HPP [13] feature sets.

From the accuracy of TU-P in Table 2, we can also found that, when  $B_2$  is fixed, lower  $B_1$  will result in significantly better detection performance, and when  $B_1$  is fixed, higher  $B_2$  will lead to a boost in detection accuracy by and large. This phenomenon is also true when comes to TU-I and TU. It can be interpreted from the following aspects. Firstly, when  $B_2$  is fixed, the single compressed video dataset is fixed and the double compressed video dataset is changing. Lower  $B_1$  may lead to larger quantization step in the process of rate-distortion control. Larger quantization step means more



**Table 5** Detection accuracy of the double compressed video fragments from its corresponding single compressed ones using TU-IP, HPP feature, and TUHPP feature on CIF video dataset

$B_1 - B_2$ (Kbps)	100–200	100–300	100–400	200–300	200–400	300–400
TU-IP(10D)						
MEAN	0.90	0.95	0.97	0.87	0.90	0.81
STDEV	0.08	0.07	0.05	0.08	0.08	0.09
MED	0.93	1.00	1.00	0.86	0.93	0.79
MAD	0.07	0.00	0.00	0.07	0.07	0.07
HPP(25D)						
MEAN	0.96	0.99	0.99	0.85	0.98	0.91
STDEV	0.05	0.03	0.02	0.09	0.03	0.07
MED	1.00	1.00	1.00	0.86	1.00	0.93
MAD	0.00	0.00	0.00	0.07	0.00	0.07
TUHPP(35D)						
MEAN	0.98	1.00	1.00	0.96	0.99	0.95
STDEV	0.04	0.00	0.00	0.05	0.02	0.05
MED	1.00	1.00	1.00	1.00	1.00	0.93
MAD	0.00	0.00	0.00	0.00	0.00	0.04

image details are lost in the first compression, and it will be easier to be detected after double compression. Secondly, when  $B_1$  is fixed, the quantity of lost image details in the first compression is fixed, but both the single and the double compressed video dataset are changing with respect to  $B_2$ . Higher  $B_2$  may lead to smaller quantization step in the recompressing process, and the hints for lost image details in the first compression may be retained better. So, higher  $B_2$  may lead to a boost in detection accuracy. Nevertheless, both the single and the double compression video databases are varying, and the

performance gain by higher  $B_2$  when  $B_1$  is fixed is not as large as the case of lower  $B_1$  when  $B_2$  is fixed.

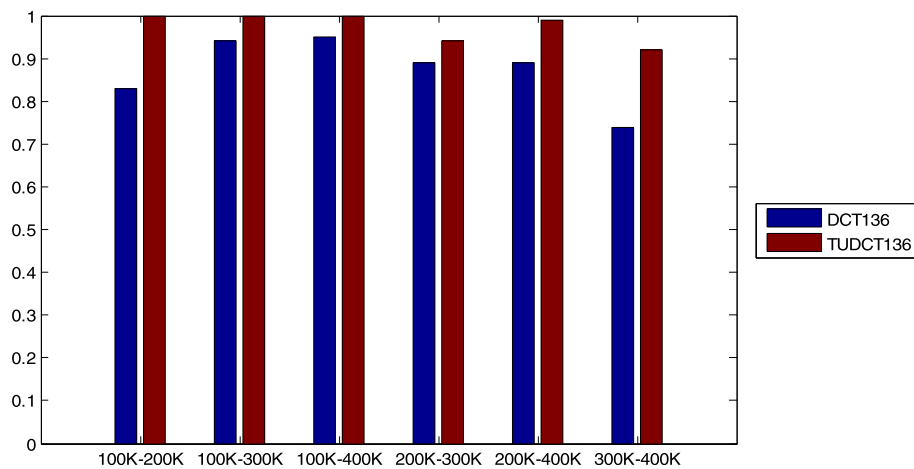
To make it easy for interested readers to reproduce the experimental results, the parameters used for training libSVM for feature set TU-I, TU-P, and TU-IP are shown in Table 3.

#### 4.3 Combination with other feature sets

TU feature is extracted from TU partition type, which is the syntactic structure of HEVC. In this subsection, we will show that TU feature set can be combined with

**Table 6** Detection accuracy of the double compressed video fragments from its corresponding single compressed ones using DCT136 [11] feature and TUDCT136 feature on QCIF video dataset

$B_1 - B_2$ (Kbps)	100–200	100–300	100–400	200–300	200–400	300–400
DCT136(136D)						
MEAN	0.83	0.94	0.95	0.89	0.89	0.74
STDEV	0.14	0.06	0.05	0.07	0.07	0.08
MED	0.88	0.97	0.96	0.89	0.89	0.75
MAD	0.04	0.03	0.04	0.06	0.06	0.07
TUDCT136(146D)						
MEAN	1.00	1.00	1.00	1.00	0.99	0.96
STDEV	0.00	0.01	0.00	0.03	0.03	0.06
MED	1.00	1.00	1.00	1.00	1.00	1.00
MAD	0.00	0.00	0.00	0.00	0.00	0.00



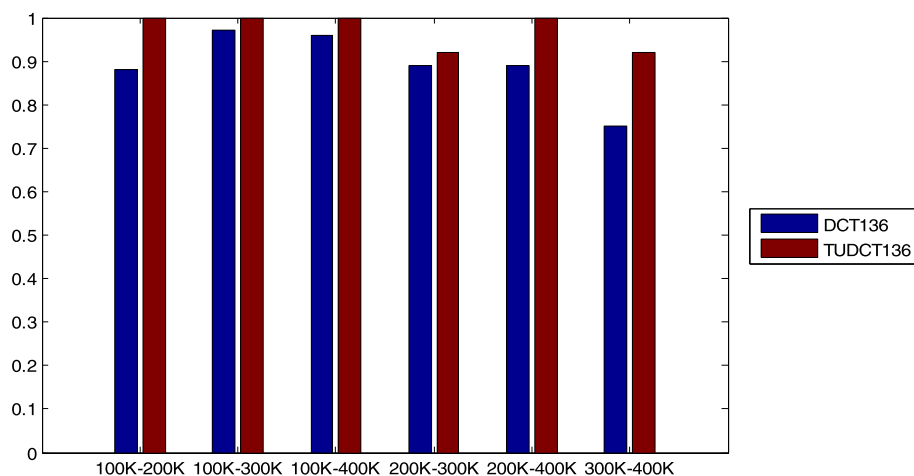
**Fig. 10** The mean value (MEAN) of detection accuracy of DCT136 [10] and TUDCT136

other feature set to boost their performance in detecting double HEVC compression.

Table 4 shows the detection accuracy of HPP [15] and HPP combined with TU (TUHPP) on QCIF video dataset. Firstly, the mean value of detection accuracy (MEAN) of HPP ranges from 0.91 to 0.99, and that of TUHPP ranges from 0.95 to 1. Figures 8 and 9 show the MEAN and MED of HPP and TUHPP on QCIF dataset. It is obvious that the combination of TU feature and HPP outperforms HPP under every double compression bitrate groups. Table 5 shows the detection accuracy of TU-IP, HPP, and HPP combined with TU (TUHPP) on CIF video dataset. And we can also observe that TUHPP outperforms HPP. TU feature is extracted from the TU partition type, while HPP feature is extracted from the PU partition type. These two kinds of feature can be a good supplement to each other.

Table 6 shows the detection accuracy of DCT136 [11] and DCT136 combined with TU (TUDCT136). Firstly, the mean value of detection accuracy (MEAN) of DCT136 ranges from 0.74 to 0.95, and that of TUDCT136 ranges from 0.92 to 1. Secondly, the MED of DCT136 lies in [0.75, 0.97], and that of TUDCT136 lies in [0.92, 1]. Meanwhile, we can observe from Figs. 10 and 11 that the combination of TU feature and DCT136 outperforms DCT136 under every double compression bitrate groups. TU feature is extracted from TU partition type, which is the syntactic structure of HEVC. DCT136 is extracted from the DCT coefficients of a video, which is the content data of a video. The combination of these two kinds of feature can boost the performance of each other.

To make it easy for interested readers to reproduce the experimental results, the parameters used for



**Fig. 11** The median value (MED) of detection accuracy of DCT136 [10] and TUDCT136

**Table 7** The parameters used for training libSVM for feature set HPP [15], TUHPP, DCT136 [11], and TUDCT136 on QCIF video dataset

$B_1 - B_2$ (Kbps)	100–200	100–300	100–400	200–300	200–400	300–400
HPP(25D)						
c	1.4142	2.0000	0.7071	2.0000	2.0000	2.0000
g	0.5000	0.3536	1.4142	0.7071	0.7071	0.5000
TUHPP(35D)						
c	0.5000	0.5000	0.7071	16	4.0000	5.6569
g	0.3536	0.5000	0.7071	0.1768	0.5000	0.1768
DCT136(136D)						
c	0.5000	32.0000	32.0000	32.0000	32.00	32.0000
g	8.0000	0.0313	0.0313	0.0313	0.0625	0.0625
TUDCT136(146D)						
c	0.25	0.25	0.25	1.4142	0.25	16
g	0.13	0.06	0.09	0.25	0.71	0.25

training libSVM for feature set HPP [15], TUHPP, DCT136 [11], and TUDCT136 on QCIF video dataset are shown in Table 7, and the parameters for TU-IP, HPP, and TUHPP on CIF dataset are shown in Table 8.

## 5 Conclusions

In this paper, we propose a new method to detect HEVC double compression under different bitrates. The distinguishing feature set is composed of the histogram of TU partition type. It is firstly reported in this paper to employ TU partition type-based feature to detect HEVC double compression, and the experimental results suggests its efficiency.

The detection methods proposed in ref. [11] employed the characteristic that DCT coefficients would be changed during recompression. It is a traditional kind of method utilized for double compression in video coding standard preceding HEVC. Nevertheless, HEVC brings in unique syntax units, such as CTU, PU, and TU. To the best of our knowledge, only PU number was

investigated in ref. [12–15] for detecting HEVC double compression. The effectiveness of PU number in detecting HEVC double compression motivates us to explore another unique feature, i.e., TU.

When using histogram of TU partition type as features for classification, the accuracy is above 0.84, even reaching 0.97 when distinguishing singly compressed videos with bitrate 400 Kbps from double compressed videos with bitrate (100–400) Kbps. The reason is that TU partition type is controlled by the rate-distortion optimization process, and it is sensitive to bitrate. Different bitrate may result in different TU partition type.

Our TU features are based on the syntactic structure of HEVC. When combined with DCT coefficients based features (e.g., ref. [11]), it can boost the original performance. That mainly lies in the fact that they captured characteristics of videos from different aspects. When combined with features from other kinds of syntactic structures of HEVC (e.g., HPP [14]), it can also improve the original performance.

**Table 8** The parameters used for training libSVM for feature set HPP [15], TUHPP, DCT136 [11], and TUDCT136 on CIF video dataset

$B_1 - B_2$ (Kbps)	100–200	100–300	100–400	200–300	200–400	300–400
TU-IP(10D)						
c	0.5	2.8284	1	11.3137	8	8
g	16	16	16	2.8284	2.8284	2.8284
HPP(25D)						
c	2.8284	0.25	2	0.70711	1.4142	4
g	2	0.5	1.4142	2.8284	1.4142	1
TUHPP(35D)						
c	1	0.25	1	2.8284	0.70711	5.6569
g	0.70711	0.17678	1.4142	1	0.35355	0.5

Other than borrowing ideas from double compression in other video standards and develop new methods that are universal to all/several kinds of video standards, researchers can also put some efforts on digging methods from the unique characteristic of HEVC. Apart from the PU number used in ref. [12–15] and TU partition types employed in this paper, there are many other interesting and promising characteristics of HEVC, such as the inter and intra prediction mode of PU and the merge type of PU in P frames. Our future work will be focused on utilizing these unique characteristics of HEVC to detect double compression. Meanwhile, the application of emerging new techniques in related area may boost the development of HEVC double compression detection [22, 23].

#### Abbreviations

CTU: Coding tree unit; DCT: Discrete cosine transform; HD: High definition; HEVC: High-efficiency video coding; PU: Prediction unit; QP: Quantization parameters; SN-PUPM: Prediction unit of its prediction mode; TU: Transform unit

#### Acknowledgements

The authors would like to thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

#### Authors' contributions

All authors take part in the discussion of the work described in this paper. ZL, ZZ, and LY conceived and designed the experiments. ZZ performed the experiments. ZL, ZZ, and LY analyzed the data. LY, ZZ, and GC wrote the paper. All authors read and approved the final version of the manuscript.

#### Funding

The work is founded by the National Natural Science Foundation of China (No. 61702034, No. 61401408) and Beijing municipal education commission project (No.KM201510015010).

#### Availability of data and materials

We can provide the data.

#### Competing interests

The authors declare that they have no competing interests.

#### Author details

<sup>1</sup>Department of Information Engineering, Beijing Institute of Graphic Communication, Beijing 102600, China. <sup>2</sup>School of Computer Science and Engineering, Beihang University, Beijing 100083, China. <sup>3</sup>School of Electronic and Information Engineering, Beijing Jiaotong University, Beijing 100044, China. <sup>4</sup>School of Computer Science, Communication University of China, Beijing 100024, China.

Received: 1 November 2018 Accepted: 16 May 2019

Published online: 29 May 2019

#### References

1. X. Guo, X. Cao, X. Chen, Y. Ma, in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*. Video editing with temporal, spatial and appearance consistency (2013), pp. 2283–2290
2. P. Bestagini, K.M. Fontani, S. Milani, M. Barni, A. Piva, M. Tagliasacchi, K.S. Tubaro, in *Proc. of Asian-Pacific Signal and Information Processing Association. An Overview on Video Forensics* (2012), pp. 1229–1233
3. G.J. Sullivan, J. Ohm, W.J. Han, T. Wiegand, Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. Circuits Syst. Video Technol.* **22**(12), 1649–1668 (2012)
4. W. Chen, Y.Q. Shi, in *Digital Watermarking*. Detection of double MPEG compression based on first digit statistics (Springer, Berlin, 2008), pp. 16–30
5. W. Luo, M. Wu, J. Huang, in *Proc. of SPIE*. MPEG recompression detection based on block artifacts, vol 6819 (2008), pp. 68190–68112
6. X. Jiang, W. Wang, T. Sun, Y.Q. Shi, S. Wang, Detection of double compression in MPEG-4 videos based on Markov statistics. *IEEE Signal Processing Lett.* **20**(5), 447–450 (2013)
7. W. Wang, X. Jiang, S. Wang, T. Sun, in *Proc. of Visual Communications and Image Processing*. Estimation of the primary quantization parameter in MPEG videos (2013)
8. D. Liao, R. Yang, H. Liu, "Double H.264/AVC compression detection using quantized nonzero AC coefficients", in *Proc. of SPIE - The International Society for Optical Engineering*, vol. 7880, 2, pp. 78800–78810, 2011
9. J. Hou, Z. Zhang, Y. Zhang, J. Ye, Y.Q. Shi, Detecting multiple H.264/AVC compressions with the same quantization parameters. *IET Inf. Secur.* **11**(3) (2016). <https://doi.org/10.1049/iet-ifs.2015.0361>
10. X. Jiang, P. He, T. Sun, F. Xie, S. Wang, Detection of double compression with the same coding parameters based on quality degradation mechanism analysis. *IEEE Trans. Inf. Forensics Secur.* **13**(1), 170–185 (2018)
11. M. Huang, R. Wang, J. Xu, D. Xu, Q. Li, in *Proc. of International Workshop on Digital Watermarking*. Detection of double compression for HEVC videos based on the co-occurrence matrix of DCT coefficients (2015), pp. 61–71
12. R. Jia, Z. Li, Z. Zhang, D. Li, "Double HEVC compression detection with the same QPs based on the PU number," in *Proc. Of 3rd Annual International Conference on Information Technology and Applications*, Vol. 02010, 7, pp. 1–4, 2016
13. Q. Xu, T. Sun, X. Jiang, Y. Dong, in *Proc. of International Workshop on Digital Watermarking*. HEVC double compression detection based on SN-PUPM feature (2017), pp. 3–17
14. Z.-H. Li, R.-S. Jia, Z.-Z. Zhang, X.-Y. Liang, J.-W. Wang, in *Proc. ITM Web Conf.* Double HEVC compression detection with different bitrates based on co-occurrence matrix of PU types and DCT coefficients, vol 12 (2017), p. 01020
15. X. Liang, Z. Li, Y. Yang, Z. Zhang, Y. Zhang, Detection of double compression for HEVC videos with fake bitrate. *IEEE Access* **6**, 53243–53253 (2018)
16. V. Sze, M. Budagavi, G.J. Sullivan, *High efficiency video coding (HEVC). Integrated Circuit and Systems, Algorithms and Architectures*, (Springer, 2014), pp. 1–375
17. <https://github.com/lheric/GitHEVCAnalyzer>, Accessed 2 Aug 2017
18. <http://www.media.xiph.org/video/derf/>, Accessed 2 Aug 2015
19. <http://www.trace.eas.asu.edu/yuv/index.html>, Accessed 2 Aug 2015
20. <http://download.csdn.net/download/amyamayadi/7903385>, Accessed 2 Aug 2015
21. C.C. Chang, C.J. Lin, in *ACM Transactions on Intelligent Systems and Technology*. LIBSVM : a library for support vector machines, vol 2 (2011), pp. 1–27 Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm> (Accessed 14 Dec 2017)
22. C. Yan, L. Li, C. Zhang, B. Liu, Y. Zhang, Q. Dai, Cross-modality bridging and knowledge transferring for image understanding. *IEEE Trans. Multimedia*. pp. 1–1 (2019)
23. C. Yan, H. Xie, J. Chen, Z. Zha, X. Hao, Y. Zhang, Q. Dai, A fast Uyghur text detector for complex background images. *IEEE Trans. Multimedia*. **20**(12), 3389–3398 (2018).

## 6 Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:**

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)