International Journal of Security and Its Application
Vol.11, No.1 (2017), pp.149-164
http://dx.doi.org/10.14257/ijsia.2017.11.1.13

# Video Authentication against Set of Temporal Tampering

Manish K Thakur[1], Vikas Saxena[1] and J P Gupta[2]

[1] *Jaypee Institute of Information Technology, Noida, India*
[2] *Hydrocarbons Education and Research Society, New Delhi, India*
*mthakur.jiit@gmail.com, vikas.saxena@jiit.ac.in, jaip.gupta@gmail.com*

***Abstract***

*In last decade, we have seen tremendous increment in video based forgery due to advancements in video editing technology and easy availability of videos. Due to sensitiveness of submitted evidences, court proceedings are mostly affected by these forged or doctored videos and thus forensic experts need to examine authenticity of visual evidences by detecting tampering if any. One of the major challenges before forensic experts is to examine the authenticity of video evidences without having additional information (i.e. passive or no reference tamper detection) viz. source of capturing device or embedded marks. This paper presents set of no reference algorithms which examine authenticity of video evidences and identify the location of tampering in videos, if videos are tampered by frame drop (or removal) and frame (scene) copying. Proposed schemes efficiently identify locations of tampering in videos, where, the observed accuracies to identify the location of tampering, if video is tempered by frame copying is in between 73.33% and 100%, whereas, it is in between 82.42% and 88.38% to detect the location of tampering of frame drop.*

***Keywords****: video authentication; video tampering; frame drop; frame copy; scene change.*

## 1. Introduction

With the advancement in visuals technology and easy availability of video editing tools, videos publicly available at many video sharing websites (like YouTube) or videos captured by CCTV can easily be manipulated (or doctored) before getting it produced as evidence during court trials, consequently may lead to misguide court proceeding [1-3]. Therefore, in practice whenever a video sequence is produced, its integrality and authenticity is to be examined (by detecting tampering – if any as well as location of tampering) before its consideration as evidence during court trials, *i.e.* videos presented as evidences need to be authenticated [4-6].

With the help of forensic experts and tools, forensic laboratories play vital role to examine these video sequences against tampering and ensure the authenticity. Depending upon availability of reference, forensic experts generally examine video sequences in three different modes *viz.* full reference, FR; reduced reference, RR; and no reference, NR [6].

While tampering detection, forensic experts may have prior information (*i.e.* FR or RR) about video contents in terms of either some embedded marks (usually watermarks or digital signatures) or detailing about video capturing device. Recently many watermarking based schemes have been proposed by researchers which authenticate a video sequence against various attacks and tampering [7-9].

Unlike FR or RR (often called as active tampering detection techniques), while tampering detection under NR mode (often called as blind or passive tampering detection), forensic experts do not have any information about actual contents of video sequence to be examined [10, 11]. Compared to FR or RR, blind tampering detection (NR

ISSN: 1738-9976 IJSIA

mode) is relatively new research focus and puts lot of challenges against forensic experts to detect tampering in manipulated video sequences.

Further, an attacker can tamper a video sequence by manipulating it spatially or temporally, thus creates a tampered (or doctored) video sequence. Spatial tampering refers to the manipulations at intra-frame level whereas temporal tampering is performed at inter-frame level where manipulations are carried with the order of frames.

As presented in Figure 1, some of the possible temporal manipulations are: drop or removal of frames (frame drop); reordering of the video frames (frame swapping); and addition of extra frames (frame copying). These temporal manipulations can be performed by dropping or swapping or adding (copying) video frames on sequence of frame level or scene level [12-18].

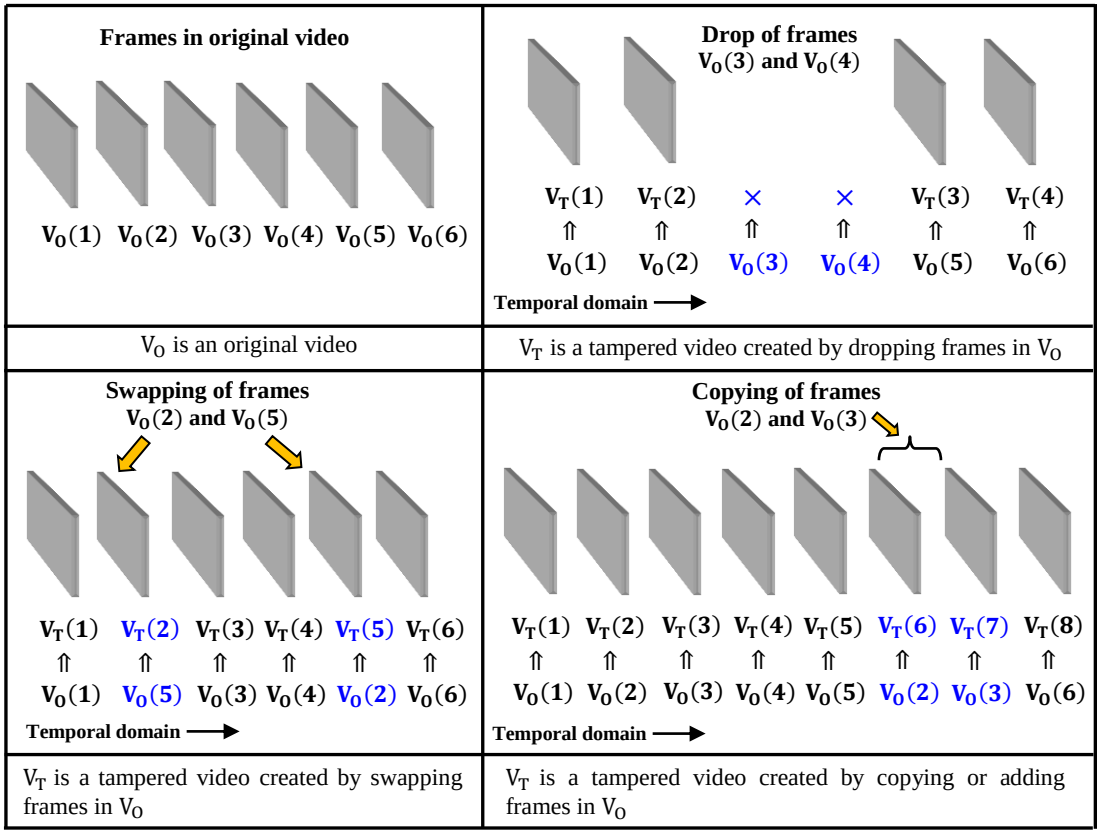



**Figure 1. Examples of Common Temporal Tampering**

Further, as an attacker can manipulate a video sequence in multiple ways, the problem of tampering detection under NR mode seems to be intractable in nature. Because of the severity and complex nature of the problem, it still put forth as current challenge before scientific community and seeks immediate attention. Therefore, in this paper we propose set of schemes for tampering detection under NR mode if the video sequences to be presented as evidence during court trials are tampered either by frame drop or frame copying.

Apart from introduction, remaining paper is organized as follows: Section 2 proposes two schemes which are used as pre-processing steps in subsequent sections *viz.* change of scene detection and count of displaced blocks. Thresholds and their recommended values used in this paper are also presented in this section. Section 3 and Section 4 present schemes for tampering detection in tampered video sequences due to frame copying and frame drop respectively, followed by conclusion and references.

## 2. Pre-processing

This section presents two schemes *viz.* scene change detection and count of displaced blocks, which are used as pre-processing steps in schemes presented in next sections. Apart from these schemes, we used some static thresholds and margin parameters in this paper. These thresholds and their recommended values are presented in Section 2.3.

### 2.1. Scene Change Detection

Scene change detection in a video sequence is one of the pre-processing steps used in the schemes presented in next sections. Depending upon amount of change between two consecutive video frames, in literature, change of scene (COS) in a video sequence has been described in different ways; some of them are as follows: significant change in between two consecutive frames, and abrupt change in between two consecutive video frames. Due to its application in many video processing applications like video compression, time to time it is being addressed by research community and handled in different ways. Threshold based schemes are most often in use where scene change is identified using various static and dynamic thresholds [19-21]. Efficiency of these schemes to identify COS depends upon recommended threshold values and matter of further discussion.

In this paper, we call there is change of scene in a given video sequence if there is abrupt change in between two consecutive video frames and propose a scheme which identifies change of scene in a given video sequence. Subsequently we present the problem statement for COS and our proposed scheme to detect COS.

#### Problem Statement

Let us given a video sequence $V_S$ with $m$ frames (a video sequence $V_S$ with 17 frames is presented in Figure 2 *i.e.* $m = 17$) and $n$ instances of abrupt changes in between two consecutive video frames (*i.e.* $n + 1$ video scenes). For the example presented in Figure 2, there are abrupt change in between video frames $V_S(5)$ and $V_S(6)$, $V_S(10)$ and $V_S(11)$, and $V_S(14)$ and $V_S(15)$ *i*.e. $n = 3$. Here, $V_S(i)$ is $i^{th}$ frame in $V_S$ *viz.* $V_S(5)$ is $5^{th}$ frame in $V_S$ which is $V_{S5}$. Our objective is to identify frame indices in $V_S$ after which there is change of scene. In the example of Figure 2, these indices are 5, 10, and 14.

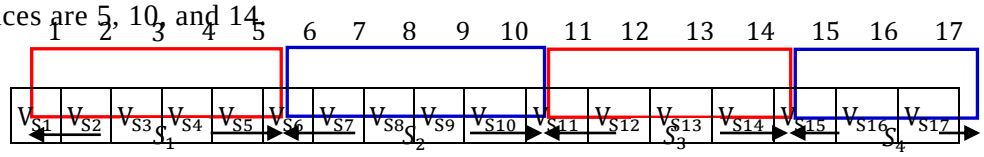



**Figure 2. A Video $V_S$ with 17 Frames ($V_S(1)$ to $V_S(17)$) and 4 Scenes ($S_1$ to $S_4$)**

#### Proposed Scheme

To identify change of scene, it is required to compare adjacent frames for similarity or dissimilarity. Many features like absolute difference, Mean Squared Error (MSE), entropy, average object area, and color histogram have been used by research community to compare video frames [14, 22]. However, performance of these features to analyze change of scene is matter of further discussion, we used MSE to compare two adjacent frames and propose a dynamic dual threshold scheme where we compare two adjacent frames (using MSE) for abrupt change.

Subsequent paragraphs describe the involved steps. Here we used various static and dynamic thresholds, which have been defined in Section 2.3, where we presented the recommended values for different static thresholds. Some thresholds have been dynamically computed in given video sequence.

***Step 1:*** For each frame $i$ in $V_S$ compute the difference by comparing it with frame $i+1$ and store it into a vector $D$.

$$i.e.\ D(i) = MSE\ (\ V_S\ (i), V_S\ (\ i+1\ )\ )\quad for\ \forall\ i = 1\ to\ m$$

where, $V_S\ (i)$ is the $i^{th}$ frame in video sequence $V_S$ and $MSE\ (\ V_S\ (i), V_S\ (\ i+1\ )\ )$ returns Mean Squared Error between $i^{th}$ and $(i+1)^{th}$ frame of $V_S$.

***Step 2:*** Compute global average, $MSE_{global}$ as average of vector $D$.

$$i.e.\ MSE_{global} = (\ \sum_{i=1}^{m} D(i))/m$$

***Step 3:*** Compute high average difference $MSE_{high}$ (which we call as dual threshold) as follows

$$MSE_{high} = \left(\frac{\sum_{i=1}^{m} D(i)}{t}\right),\quad if\ D(i)\ \geq\ MSE_{global}$$

where, $t$ is the count of $i$ for which $D(i)\ \geq\ MSE_{global}$

***Step 4:*** Store all frame indices $i$ (from 1 to $m$) of $V_S$ into a vector $SChange$ if $D(i)\ \geq\ MSE_{high}$.

***Step 5:*** Check video frames $V_S(i)$ and $V_S(i+1)$ for count of displaced blocks (using scheme $blkDisp(\ )$ presented in Section 2.2) if $D(i)$ is at the margin of $MSE_{high}$ *i.e.* $C_1\ \times MSE_{high} \leq D(i) \leq C_2 \times MSE_{high}$. For efficient identification of COS tune margin parameters $C_1$ and $C_2$.

Include frame index $i$ of $V_S$ into vector $SChange$ if $D(i) \geq\ C_1\ \times MSE_{high}$ and $blkDisp(\ V_S(i), V_S(i+1))/C\ \geq DISP_{diff}$. Discard frame $i$ if $D(i)\ \leq\ C_2\ \times MSE_{high}$ and $blkDisp(\ V_S(i), V_S(i+1))/C\ \leq\ DISP_{gradual}$. Here, $DISP_{diff}$ and $DISP_{gradual}$ are static thresholds and their recommended values are presented in Section 2.3 (Table 1), whereas $C$ is the count of $8 \times 8$ blocks in a frame of given video sequence. Repeat Step 2 to Step 6, if there is no frame left at margin either to be included or discarded as change of scene.

***Step 6*:** Report, indices in $SChange$ as frame indices in $V_S$ after which there is a change of scene.

Thus, after above steps, vector $SChange$ in the example of Figure 2 will report frame indices 5, 10, and 14 as frame indices in $V_S$ after which there is change of scene.

### 2.2. Count of Displaced Blocks

Block based computations are most often in use for various video processing like video compression. These computations include block based MSE or block based Histogram or block based entropy [22]. This section presents a scheme which computes the count of displaced blocks (of size $8 \times 8$) in two video frames. This scheme is one of the pre-processing steps used in schemes presented in Section 2.2, Section 3 and Section 4.

Here for given video frames (say $F_C$ and $F_N$), we are interested to find out count of $8 \times 8$ blocks of $F_C$ which are not present in $F_N$. A scheme is presented in Figure 3 which computes this count using static threshold $bThr$ and margin parameters $m_1$ and $m_2$ (Table 1 presents their recommended values). We used Shannon's entropy to compute information contained by $8 \times 8$ blocks of $F_C$ and $F_N$. Entropy of each block in $F_C$ is compared with blocks in $F_N$. Later probable matching blocks of $F_C$ and $F_N$ are checked with block threshold $bThr$.

**Algorithm: $blkDisp$ ($F_C$ , $F_N$)**

{Computes and return count of $8 \times 8$ blocks of $F_C$ not present in $F_N$}

$width$ , $height$ {Width and height of video frame $F_C$or $F_N$}
$col$ , $row$ {Column and row wise count of $8 \times 8$ block in a frame}
$bCount$ {Total count of $8 \times 8$ blocks in a frame *i.e.* $bCount = row \times col$}
$entC[1..bCount][1..3]$ {A matrix which stores entropy of an $8 \times 8$ block of $F_C$ and its row and col index}
$entN[1..bCount][1..3]$ {A matrix which stores entropy of an $8 \times 8$ block of $F_N$ and its row and col index}
$mark[1..bCount]$ {A vector which stores Boolean value}

begin

initialize $k \leftarrow 0, mC \leftarrow 0, row \leftarrow height/8 , col \leftarrow width/8$

for $i \leftarrow 1$ to $row$ do
for $j \leftarrow 1$ to $col$ do
$entC[k][0] \leftarrow ENT\ (F_C\ (i,j))$ , $entN[k][0] \leftarrow ENT\ (F_N\ (i,j))$
{$F_C\ (i,j)$ is $8 \times 8$ block of frame $F_C$ indexed at $i$, $j$, and $ENT$ ( ) returns entropy of that block}
$entC[k][1] \leftarrow i$ , $entC[k][2] \leftarrow j$ , $entN[k][1] \leftarrow i$ , $entN[k][2] \leftarrow j$ , $k \leftarrow k+1$
end for, end for

for $i \leftarrow 1$ to $bCount$ do
$mark[\,i\,] \leftarrow 0$

for $i \leftarrow 1$ to $bCount$ do
for $j \leftarrow 1$ to $bCount$ do
$v_1 \leftarrow m_1 \times entN[j][0]$
$v_2 \leftarrow m_2 \times entN[j][0]$ {Table 1 (Section 2.3) presents recommended values for $m_1$ and $m_2$}
if $v_1 \geq entC[i][0]$ and $v_2 \leq entC[i][0]$ then
$check \leftarrow MSE(F_C(entC[i][1], entC[i][2]), F_N(entN[j][1], entN[j][2]))$
{ $MSE$( ) returns Mean Squared Error between $8 \times 8$ block of $F_C$and $8 \times 8$ block of $F_N$}
if $check \leq bThr$ and $mark[i] = 0$ then {$bThr$ recommended value is presented in Table 1}
$mC \leftarrow mC + 1$ , $mark[i] \leftarrow 1$
end if, end if
end for, end for

return $(bCount - mC)$

**Figure 3. Algorithm $blkDisp$ Computes Count of $8 \times 8$ Blocks in Video Frame $F_C$ not Present in Frame $F_N$**

### 2.3. Thresholds and Constants

As listed subsequently, in this paper we have used various thresholds (static and dynamic) and constants. Dynamic thresholds are computed for a given video sequence (*viz.* $MSE_{high}$, $MSE_{global}$) whereas we recommended static threshold values based upon extensive analysis with thousands of video frames of uncompressed cif video sequences (*i.e.* frame size is of $352 \times 288$ pixels). Constants are generally used for considering margins of different computations.

$DISP_{gradual}$: This is one of the static thresholds used in this paper. It represents ratio between count of $8 \times 8$ blocks of one video frame (say $F_i$) not present in adjacent frame $F_{i+1}$ if there is gradual change between $F_i$ and $F_{i+1}$ and count of $8 \times 8$ blocks in a given video frame. *i.e.* there is little change from $F_i$ to $F_{i+1}$, neither this change is significant to consider a change of scene from $F_i$ to $F_{i+1}$, nor it is too less to consider both frame as same. We investigated approximately 42,000 video frames (*i.e.* 21,000 sets of $F_i$ and $F_{i+1}$) to recommend its value as presented in Table 1.

$DISP_{diff}$: Like $DISP_{gradual}$, it is also one of the static threshold and represents ratio between count of $8 \times 8$ blocks of $F_i$ not present in adjacent frame $F_{i+1}$ if there is abrupt change between $F_i$ and $F_{i+1}$ and count of $8 \times 8$ blocks in a given video frame. *i.e.* both frames are different and certainly we can call there is a change of scene (abrupt) between $F_i$ and $F_{i+1}$. We investigated approximately 20,000 sets of such $F_i$ and $F_{i+1}$ to recommend its value.

$DISP_{similar}$: This static threshold represents ratio between count of $8 \times 8$ blocks of $F_i$ not present in adjacent frame $F_{i+1}$ (if $F_i$ and $F_{i+1}$ are almost same) and count of $8 \times 8$ blocks in a given video frame. We investigated approximately 11,500 sets of such $F_i$ and $F_{i+1}$ to recommend its value.

$bThr$: It is another static threshold used in scheme $blkDisp()$. It represents the maximum Mean Squared Error between two $8 \times 8$ blocks (say $B_1$ and $B_2$) of video frames such that we can call both blocks almost identical. To recommend its value, we investigated approximately 15,000 sets of such $B_1$ and $B_2$.

$MTH_{Low}$ : It is the static threshold used in this paper. It represents the maximum Mean Squared Error between two video frames (say $F_1$ and $F_2$) such that we can call both frames nearly same (neither identical nor different). We investigated approximately 11,000 sets of such $F_1$ and $F_2$ to recommend its value as presented in Table 1.

Margin Constants: In this paper, we used $C_1$, $C_2$, $m_1$, and $m_2$ to consider margins during various operations *viz.* consideration of change of scene between two adjacent frames and consideration of blocks symmetry using entropies of two $8 \times 8$ blocks.

**Table 1. Recommended Values of Thresholds and Margin Constants**

| Thresholds and Margin Constants | Recommended Values |
|---|---|
| $DISP_{gradual}$ , $DISP_{diff}$ , and $DISP_{similar}$ | 0.52, 0.79, and 0.16 |
| $bThr$ and $MTH_{Low}$ | 60 and 524 |
| $C_1$ and $m_1$ | 0.98 |
| $C_2$ and $m_2$ | 1.03 |

## 3. Frame Copying Detection

As discussed in Section 1, frame (scene) copying (or addition) is one of the temporal tampering where an attacker may copy a scene $S_i$ (of $n$ frames) of original (or actual) video sequence $V_O$ (of $m$ frames) to another location in $V_O$, and thus creates a tampered video sequence $V_T$ (of $m + n$ frames). Copying of a video scene at different location (as first scene or last scene or in between scene $S_i$ and $S_j$) may change the context of visual information, and mislead the court proceedings if $V_T$ is presented as evidence.

Section 3.1 describes the problem statement; Section 3.2 presents a scheme for detecting the tampering of frame/scene copying and Section 3.3 discusses the simulation of presented scheme using different video sequences.

### 3.1. Problem Statement

Let us consider an original (actual) video sequence $V_O$ with $m$ frames ($m$ is 12 in the example of Figure 4) which is being tampered by copying some video frames/scene to another location in $V_O$ (scene $S_2$ is copied after scene $S_3$ in the example of Figure 5) and thus create tampered video $V_T$ with $n$ video frames ($n$ is 16 in the example of Figure 5).

Submission of video sequence $V_T$ as evidence during court trials may mislead court proceedings, therefore before considering $V_T$ as evidence, its authenticity is to be ensured and thus it is required to identify whether submitted evidence $V_T$ is being tampered by frame/scene copying or not (without having any information about $V_O$ *i.e.* NR tampering detection) and if found as tampered video sequence then objective is to identify the location of tampering (frame index after which tampering is detected).

Thus, in the example of Figure 5, it is expected to identify $V_T$ as tampered video sequence and tampering location as after index 12, *i.e.* $V_T(12)$.

### 3.2. Proposed Scheme

The problem presented in Section 3.1 needs first to identify different scenes in a video sequence $V_T$, then to be analyzed whether $V_T$ is tempered by frame (scene) copying or not. Subsequent paragraphs present step by step solution of the problem stated in Section 3.1.

***Step 1:*** Input a video sequence $V_T$ with $n$ video frames as $V_T(1)$, $V_T(2)$, ..... $V_T(n)$.
***Step 2:*** Apply change of scene algorithm (Section 2) to $V_T$ and store frame indices of each scene of $V_T$ into individual bins (say $k$ bins have been formed). Figure 6 represents such bins (or scenes) for video sequence $V_T$ of Figure 5.

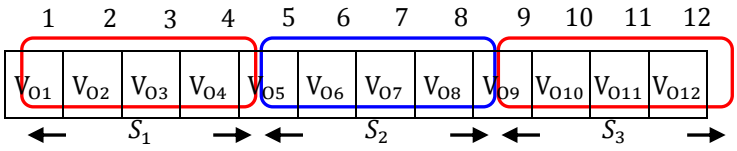



**Figure 4. Actual (Original) Video Sequence $V_0$ with Three Scenes $S_1$, $S_2$, and $S_3$**

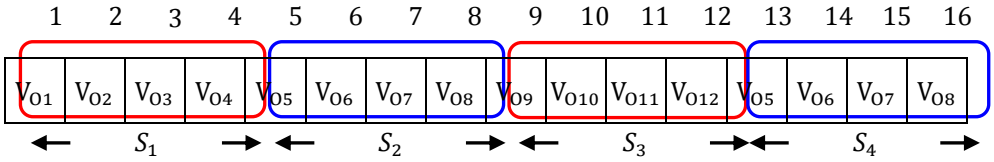



**Figure 5. Tampered video $V_T$ Created by Copying Scene $S_2$ in $V_0$ after $S_3$ in $V_0$**

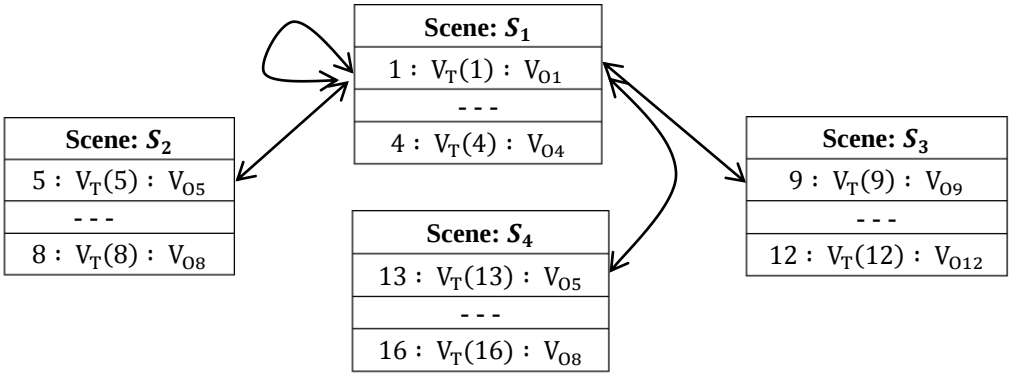



**Figure 6. Bins Containing Frame Indices of each Scene of Tampered Video $V_T$**

***Step 3:*** Compute difference between first frame of scene $S_1$ and first frame of each scene (bins) using Mean Squared Error (MSE). Store these differences into vector $SCopy$. Figure 7 presents vector $SCopy$ for the bins of Figure 6.

$$SCopy(i) = MSE\left(V_T(S_1(1)), V_T(S_i(1))\right) \quad for\ \forall\ i = 1\ to\ k$$

where, $S_i(1)$ is the $1^{st}$ frame index of scene $S_i$ and $MSE\ (a, b)$ returns MSE between frame *a* and *b*.

***Step 4:*** If $SCopy\ (i) - SCopy\ (j) < MTH_{Low}\ for\ \forall\ i,j = 1\ to\ k\ where\ i \neq j$, then check frames $V_T(S_i(1))$ and $V_T\left(S_j(1)\right)$ for similarity using scheme $blkDisp(\ )$.

***Step 5:*** If $V_T(S_i(1)) = V_T\left(S_j(1)\right)$ *i.e.* $1^{st}$ frame of scene $S_i$ and scene $S_j$ are same (*i.e* $Disp(\ )/bcount \leq DISP_{similar}$ ) then check last frames of both scenes for similarity. If last frames are also same then report $V_T$ as tampered video and location of tampering as $S_i(1)$ and $S_j(1)$. Repeat the process until all $k$ bins have been explored.

In preceding example, $SCopy\ (2)$ *i.e.* $D_1$ and $SCopy\ (4)$ *i.e.* $D_1$ are same, therefore, frames $V_T(S_2(1))$ and $V_T(S_4(1))$ have been checked for similarity and found as same. Thus, report $V_T$ as tampered video sequence and location of tampering as $S_2(1)$ and $S_4(1)$.

### 3.3. Simulation and Analysis

We simulated the scheme presented in Section 3.2 to check authenticity of video evidence (against tampering of frame copying) with 40 tampered videos and observed the performance of proposed scheme. Subsequent paragraph presents the simulation details and analysis.

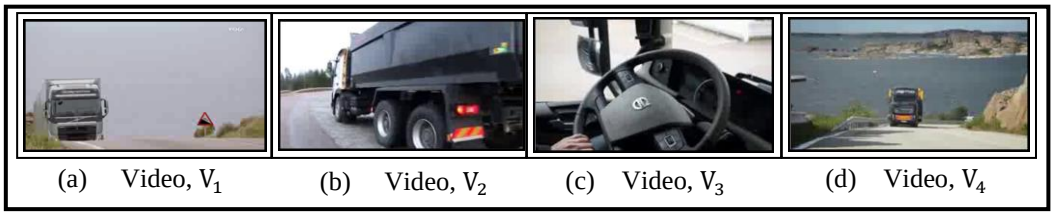
$$SCopy(1) = MSE(\mathrm{V_T}(S_1(1)), \mathrm{V_T}(S_1(1)) = MSE(\mathrm{V_{O1}}, \mathrm{V_{O1}}) = 0$$

$$SCopy(2) = MSE(\mathrm{V_T}(S_1(1)), \mathrm{V_T}(S_2(1)) = MSE(\mathrm{V_{O1}}, \mathrm{V_{O5}}) = D_1$$

$$SCopy(3) = MSE(\mathrm{V_T}(S_1(1)), \mathrm{V_T}(S_3(1)) = MSE(\mathrm{V_{O1}}, \mathrm{V_{O9}}) = D_2$$

$$SCopy(4) = MSE(\mathrm{V_T}(S_1(1)), \mathrm{V_T}(S_4(1)) = MSE(\mathrm{V_{O1}}, \mathrm{V_{O5}}) = D_1$$

**Figure 7. Vector $SCopy$ for the Bins Presented in Figure 6**



(a) Video, $V_1$ (b) Video, $V_2$ (c) Video, $V_3$ (d) Video, $V_4$

**Figure 8. Set of Videos Used as Original Videos [23-26] to Conduct Experiments**

We used 4 publicly available video sequences, $V_1$ to $V_4$ (presented in Figure 8) to simulate our scheme. We uncompressed these videos and considered them as original (actual) video sequences for our experiments. Further, we introduced the tampering of frame/scene copy by copying video scenes in a video sequence to some other location to create tampered videos. These tampered videos were created by randomly copying and pasting all the frames of 1 to 5 scenes in each original video. We created 10 tampered video sequences from each original video sequence *i.e.* in total we used 40 tampered video sequences. Table 2 consolidates the count of scenes (abrupt) in original video sequence and tampered video sequences ($T_1$ to $T_{10}$). Count of scenes in original video sequences is from 28 to 62 whereas in tampered video sequences, it ranges from 29 to 67.

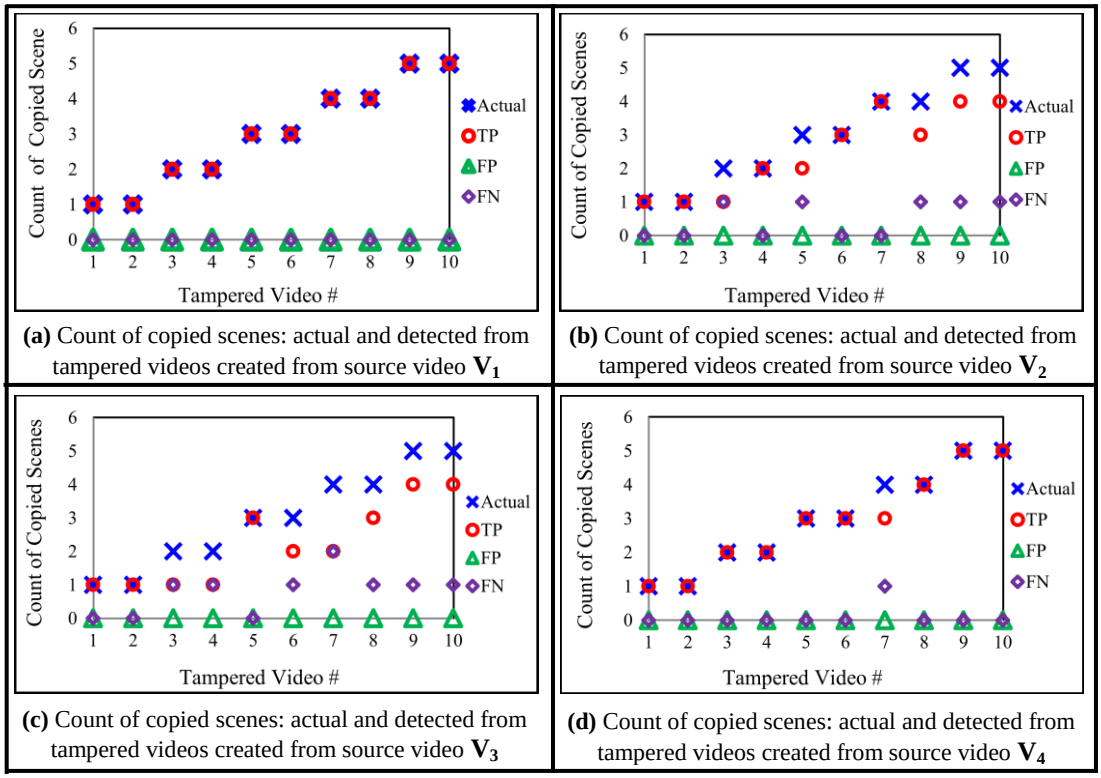
**Table 2. Count of Scenes in Original and Tampered Videos (Due to Frame copy)**

| Original Video Name | Count of scenes in original (actual) and tampered video sequences | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | Original Video | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ | $T_{10}$ |
| $V_1$ | 28 | 29 | 29 | 30 | 30 | 31 | 31 | 32 | 32 | 33 | 33 |
| $V_2$ | 62 | 63 | 63 | 64 | 64 | 65 | 65 | 66 | 66 | 67 | 67 |
| $V_3$ | 49 | 50 | 50 | 51 | 51 | 52 | 52 | 53 | 53 | 54 | 54 |
| $V_4$ | 32 | 33 | 33 | 34 | 34 | 35 | 35 | 36 | 36 | 37 | 37 |

Further, experiments were conducted with tampered videos having the objectives to identify the location of tampering and accordingly examine the authenticity of the videos. Figure 9 presents the performance of the scheme (Section 3.2) while conducting the experiments with tampered videos ($T_1$ to $T_{10}$) created from original videos $V_1$ - Figure 9 (a); $V_2$ - Figure 9 (b); $V_3$ - Figure 9 (c); and $V_4$ - Figure 9 (d). Figure 9 presents the plots between: (i) Actual count of scenes copied from original videos, ($V_1$ to $V_4$) and pasted (*i.e.*

tampering of frame copy) to other locations to create tampered videos ($T_1$ to $T_{10}$); (ii) Count of correctly detected copied frames (at scene level) in each tampered video ($T_1$ to $T_{10}$) created from original videos ($V_1$ to $V_4$). – *i.e.* True Positive (TP); (iii) Count of incorrectly detected copied frames in each tampered video ($T_1$ to $T_{10}$), created from original video ($V_1$ to $V_4$). – *i.e.* False Positive (FP); (iv) Count of incorrectly rejected copied frames in each tampered video ($T_1$ to $T_{10}$), created from original video ($V_1$ to $V_4$). – *i.e.* False Negative (FN).

As observed from Figure 9, the proposed scheme successfully identified all the 30 locations of tampering of frame copy in the tampered videos ($T_1$ to $T_{10}$) created from original video, $V_1$, whereas its performance is worst with the tampered videos created from video, $V_3$.




**(a)** Count of copied scenes: actual and detected from tampered videos created from source video $\mathbf{V_1}$

**(b)** Count of copied scenes: actual and detected from tampered videos created from source video $\mathbf{V_2}$

**(c)** Count of copied scenes: actual and detected from tampered videos created from source video $\mathbf{V_3}$

**(d)** Count of copied scenes: actual and detected from tampered videos created from source video $\mathbf{V_4}$

**Figure 9. Performance of the Scheme (Section 3.2) Involving Tampered Videos $T_1$ to $T_{10}$ Created from Original Videos, ($V_1$ to $V_4$)**

Here, our scheme could correctly identify 22 out of 30 locations of tampering of frame copy (in videos $T_1$ to $T_{10}$). The incorrectly detected (FP) and incorrectly rejected (FN) locations of tampering in the tampered videos created from $V_1$ are least, whereas it is maximum when these videos are created from $V_3$. In total, the average accuracy to correctly identify the location of tampering of frame copying is between 73.33% and 100%.

## 4. Frame Drop Detection

As discussed in Section 1, frame drop (removal or deletion) is one of the temporal tampering where an attacker may drop (remove or delete) some video frames of a scene $S_i$ (of $n$ frames) of a video sequence $V_O$ (of $m$ frames), and thus creates a tampered video sequence $V_T$ (of $m - n$ frames). Deletion of video frames in a video scene may change the context of visual information, and mislead the court proceedings if it is presented as evidence

Section 4.1 describes the problem statement; Section 4.2 presents a scheme for detecting the tampering of frame drop and Section 4.3 discusses the simulation of presented scheme.

### 4.1. Problem Statement

Let us consider an original video $V_O$ ($m$ frames), which is being tampered by dropping some frames of scene, $S_i$, thus creates a tampered video $V_T$ (less than $m$ frames). Figure 10 depicts an example of original video, $V_O$ (15 frames and 3 scenes). Further, frames $V_O(2)$ and $V_O(3)$ of scene $S_1$, frames $V_O(7)$ and $V_O(8)$ of scene $S_2$, and frames $V_O(12)$ and $V_O(13)$ of scene $S_3$ are dropped in $V_O$ to create tampered video $V_T$ (shown in Figure 11 with 9 frames).

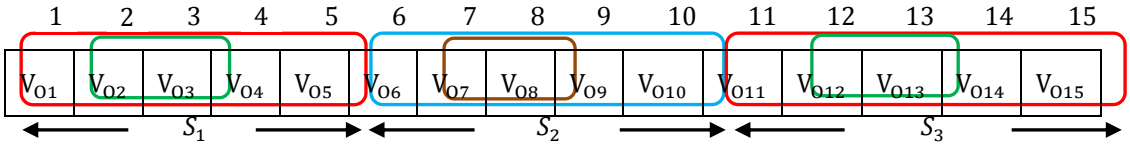



**Figure 10. An Example - Actual Original) Video Sequence $V_O$**

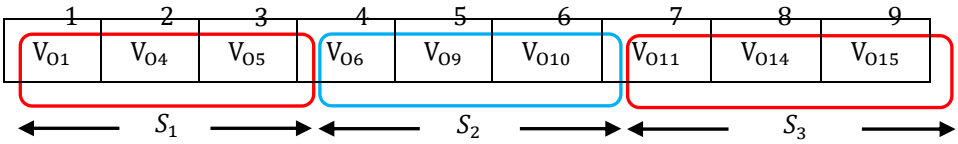



**Figure 11. Tampered Video $V_T$ is Created by Dropping Frames $V_O(2)$ to $V_O(8)$, $V_O(12)$, and $V_O(13)$ in $V_O$**

Instead of video $V_O$, submission of video $V_T$ as evidence during court trials may mislead court proceedings. Therefore before considering $V_T$ as evidence, its authenticity is to be ensured. Thus it is required to identify whether, video $V_T$ is being tampered by frame drop or not (without any information about $V_O$ *i.e.* NR tampering detection) and if found as tampered then identify the location of tampering (frame index after which tampering is detected).

Thus in the example of Figure 11, it is expected to identify $V_T$ as tampered video sequence and tampering location as after frame indices 1, 6, and 11.

### 4.2. Proposed Scheme

The problem presented in Section 4.1 needs first to identify different scenes in a video sequence $V_T$, then only it can be analysed whether $V_T$ is tempered by frame drop or not. Subsequent paragraphs present step by step solution of the problem stated in Section 4.1.

***Step 1:*** Input a video sequence $V_T$ with n video frames as $V_T(1)$, $V_T(2)$, ..... $V_T$(n

***Step 2:*** Apply change of scene algorithm (Section 2) to $V_T$ and store frame indices of each scene of $V_T$ into individual bins (say $k$ bins). Figure 12 represents such bins (or scenes) for video sequence $V_T$ of Figure 11

***Step 3:*** For each bin (or scene), compute difference $D(i,j)$ of adjacent frames by comparing $j^{th}$ and $(j+1)^{th}$ frames of scene $S_i$ using Mean Squared Error.

$$D(i,j) = MSE\left(V_T(S_i(j)), V_T(S_i(j+1))\right), \quad for\ \forall\ i = 1\ to\ k\ \ and\ \ \forall\ j = 1\ to\ fcount(S_i)$$

where, $S_i(j)$ is the index of $j^{th}$ frame in $S_i$, and $fcount(S_i)$ returns count of frames in $S_i$.

| Scene: $S_1$ |
|---|
| $1 : V_T(1) : V_{O1}$ |
| $2 : V_T(2) : V_{O4}$ |
| $3 : V_T(3) : V_{O5}$ |

| Scene: $S_2$ |
|---|
| $4 : V_T(4) : V_{O6}$ |
| $5 : V_T(5) : V_{O9}$ |
| $6 : V_T(6) : V_{O10}$ |

| Scene: $S_3$ |
|---|
| $7 : V_T(7) : V_{O11}$ |
| $8 : V_T(8) : V_{O14}$ |
| $9 : V_T(9) : V_{O15}$ |

**Figure 12. Bins Containing Frame Indices of each Scene of Tampered Video $\mathbf{V_T}$**

***Step 4***: For each bin $i$ (from 1 to $k$), do following

(a) Compute average Mean squared Error ($AMSE$) for bin $i$ as follows

$$AMSE(i) = \frac{\sum_{j=1}^{fcount(S_i)} D(i,\ j)}{fcount(S_i)}$$

(b) Compute high Mean Squared Error ($HMSE$) for bin $i$ as follows

$$HMSE(i) = \frac{\sum_{j=1}^{fcount(S_i)} D(i,\ j)}{t}, \quad if\ D(i,j)\ \geq AMSE(i)$$

(c) Use $blkDisp(\ )$ scheme (Section 2.2) to compute count of displaced block $BD(i,j)$ between frames in $V_T$ indexed at $S_i(j)$ and $S_i(j+1)$ of bin $i$ if $D(i,j)\ \geq HMSE(i)$.

$$BD(i,j) = blkDisp\left(V_T(S_i(j)), V_T(S_i(j))\right), \quad if\ D(i,j) \geq HMSE(i)$$

(d) Report there is tampering of frame drop after frame index $S_i(j)$, if

$BD(i,j)/bcount\ \geq\ DISP_{gradual}$, where $bcount$ is count of $8 \times 8$ blocks in a frame

***Step 5***: Report $V_T$ as tampered video sequence if there is reporting of tampering in Step 4d.

In the preceding example, there might be possibility that $D(1,1)$ is greater than high mean square error and further, $BD(1,1)\ \geq\ DISP_{gradual}$, thus report $S_1(1)$ as frame indices in $V_T$ after which there is a tampering of frame drop. Similarly other frame indices can also be obtained as frame indices after which there is a tampering of frame drop.

### 4.3. Simulation and Analysis

We simulated the scheme presented in Section 4.2 to examine the authenticity of videos against tampering of frame drop. Subsequently we present the simulation details and analysis.

We used 4 publicly available uncompressed video sequences (Figure 8) to simulate our scheme. We created 15 copies of each original video and introduced the tampering of frame drop by randomly deleting some frames (5 to 15 frames) of video scenes in these copies of original videos, i.e. we created 15 tampered video sequences ($T_1$ to $T_{10}$) from each original video sequence. In total we used 60 tampered video sequences to conduct the experiments.

Table 3 details the count of sets of dropped frames (where one set of dropped frames represent deletion of 5 to 15 intermediate frames in a video scene in the original video) in each tampered videos, *i.e.* $T_1$ to $T_{15}$ created from copies of original videos, $V_i$, *viz.* we

dropped 10 sets of intermediate frames in the copy of original video, $V_1$ to create the tampered video, $T_1$.

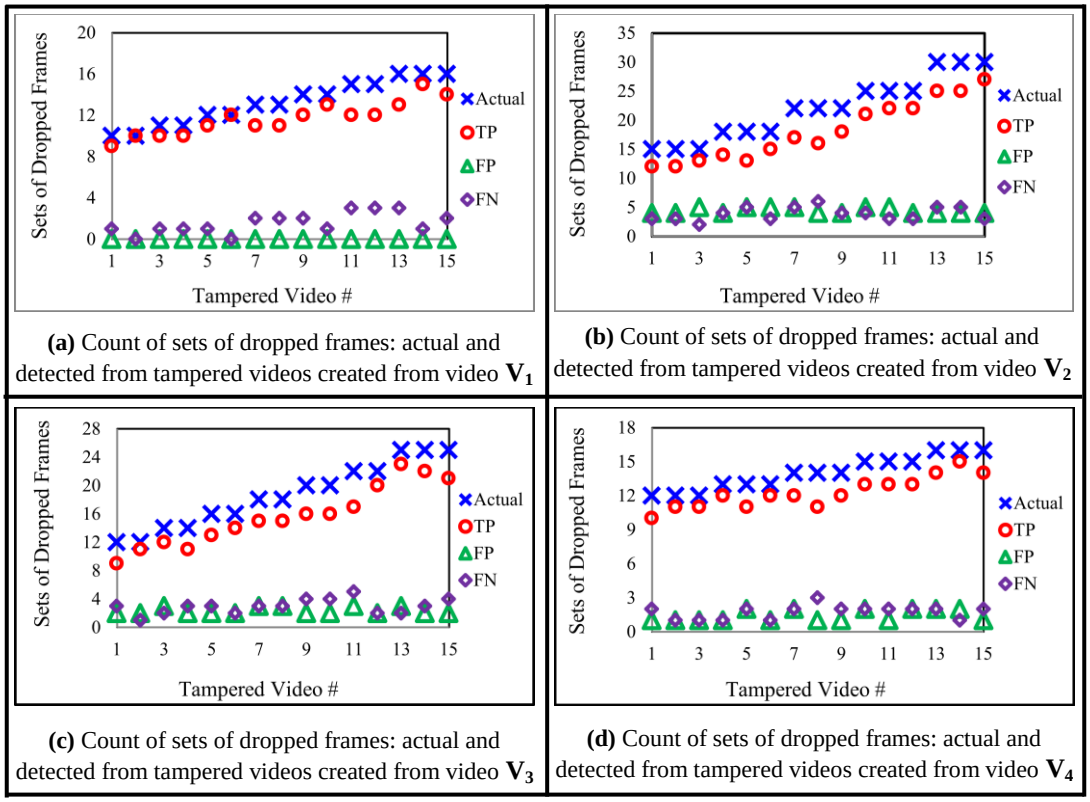
In total we dropped 198 sets of intermediate frames from 198 scenes of video, $V_1$ to create 15 tampered videos ($T_1$ to $T_{15}$). This count is 330, 279, and 210 for $V_2$, $V_3$, and $V_4$ respectively. In all, we dropped 1017 sets of intermediate frames to create 60 tampered videos.

We conducted experiments with the help of these 60 tampered videos to analyze the performance of the proposed scheme (Section 4.2), where the objectives are to examine the authenticity of videos and location of tampering of frame drop, if video is found as tampered.

**Table 3. Actual Count of Sets of Dropped Frames in Tampered Videos**

| **Original Videos** | | **Actual count of sets of dropped frames in each tampered videos (*viz.* $T_1$ to $T_{15}$) created from respective original videos** | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Video Name | Count of Scenes | $T_1$ | $T_2$ | $T_3$ | $T_4$ | $T_5$ | $T_6$ | $T_7$ | $T_8$ | $T_9$ | $T_{10}$ | $T_{11}$ | $T_{12}$ | $T_{13}$ | $T_{14}$ | $T_{15}$ |
| $V_1$ | 28 | 10 | 10 | 11 | 11 | 12 | 12 | 13 | 13 | 14 | 14 | 15 | 15 | 16 | 16 | 16 |
| $V_2$ | 62 | 15 | 15 | 15 | 18 | 18 | 18 | 22 | 22 | 22 | 25 | 25 | 25 | 30 | 30 | 30 |
| $V_3$ | 49 | 12 | 12 | 14 | 14 | 16 | 16 | 18 | 18 | 20 | 20 | 22 | 22 | 25 | 25 | 25 |
| $V_4$ | 32 | 12 | 12 | 12 | 13 | 13 | 13 | 14 | 14 | 14 | 15 | 15 | 15 | 16 | 16 | 16 |

Figure 13 presents the performance of the proposed scheme while conducting the experiments with the tampered videos, $T_1$ to $T_{15}$ created from copies of original videos, $V_i$ ($V_1$ to $V_4$). Figure 13 presents the plot between, (i) Actual count of sets of dropped frames which have been dropped from some scenes in the copies of original videos, $V_1$ to $V_4$; (ii) count of correctly detected sets of dropped frames in each tampered video ($T_1$ to $T_{15}$) created from copies of original videos $V_1$ to $V_4$, *i.e.* True Positive (TP); (iii) count of incorrectly detected sets of dropped frames in each tampered video created from copies of original videos $V_1$ to $V_4$, *i.e.* False Positive (FP); (iv) Count of incorrectly rejected sets of dropped frames in each tampered video created from copies of original videos, $V_1$ to $V_4$, *i.e.* False Negative (FN).




**(a)** Count of sets of dropped frames: actual and detected from tampered videos created from video $\mathbf{V_1}$

**(b)** Count of sets of dropped frames: actual and detected from tampered videos created from video $\mathbf{V_2}$

**(c)** Count of sets of dropped frames: actual and detected from tampered videos created from video $\mathbf{V_3}$

**(d)** Count of sets of dropped frames: actual and detected from tampered videos created from video $\mathbf{V_4}$

**Figure 13. Performance of the Scheme (Section 4.2) Involving Tampered Videos $T_1$ to $T_{15}$ Created from Original Videos, ($V_1$ to $V_4$)**

As observed from Figure 13 (a), the proposed scheme successfully identified 175 locations as set of dropped frames (the actual count was 198) in the tampered videos ($T_1$ to $T_{15}$) created from original video, $V_1$, whereas its performance is worst with the tampered videos created from video, $V_2$. Here, out of 330 locations of tampering of frame drop, our scheme correctly identify 272 such locations. Further, the incorrectly detected (FP) and incorrectly rejected (FN) locations of tampering in the tampered videos created from $V_1$ are least, whereas it is maximum when tampered videos are created from $V_2$. In total, the average accuracy to correctly identify the location of tampering of frame drop is between 82.42% and 88.38%.

## 5. Conclusion

In this paper we presented set of tampering detection schemes *viz.* frame (or scene) copying detection and frame drop (removal or deletion) detection, which examine the authenticity of video sequences submitted as evidence during court trials. In the process, we also proposed schemes for scene change detection and count of displaced blocks which are used as pre-processing steps in the schemes proposed for tampering detection. Proposed schemes efficiently identify the tampering (if any) in a video sequence without having any information about its original (actual) contents. Altogether we observed accuracies in between 73.33% and 100% for detection of the tampering location of frame copying (at scene level) and in between 82.42% and 88.38% for detection of the tampering location of frame drop.

## References

[1] http://www.bbc.co.uk/news/science-environment-20629671/. Accessed 30 Jan 2014
[2] http://www.videoforensicexpert.com/tag/digital-video-forensic-evidence/. Accessed 30 Jan 2014
[3] http://www.videoforensicexpert.com/video-forensics/video-authentication-services/. Accessed 30 Jan 2014
[4] A. Rocha, W. Scheirer, T. Boult, and S. Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Computing Surveys*, Vol. 43, No. 4, Article 26, (**2011**), pp. 1-42.
[5] J. A. Redi, W. Taktak, and J. L. Dugelay, "Digital image forensics: a booklet for beginners," *Multimed Tools Appl*, Vol. 51, Issue 1, (**2011**), pp. 133–162.
[6] M. K. Thakur, V. Saxena, J. P. Gupta, "Data-parallel full reference algorithm for dropped frame identification in uncompressed video using genetic algorithm," in *Proc. 6th International Conference on Contemporary Computing* (IC3 2013), August 8-10, (**2013**), pp. 467-471
[7] S. D. Roy, X. Li, Y. Shoshan, A. Fish, and O. Yadid-Pecht, "Hardware Implementation of a Digital Watermarking System for Video Authentication," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 23, No. 2, (**2013**), pp. 289-301
[8] S. Chen and H. Leung, "Chaotic Watermarking for Video Authentication in Surveillance Applications," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 18, No. 5, May 2008, pp. 704-709
[9] P. K. Atrey, W. Q. Yan, and M. S. Kankanhalli, "A scalable signature scheme for video authentication," *Multimed Tools Appl*, 34, (**2007**), pp. 107–135
[10] T. C. Lin – I, C. Min-Kuan, and C. You-Lin, "A Passive-Blind Forgery Detection Scheme Based on Content-Adaptive Quantization Table Estimation," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 21, Issue 4, (**2011**), pp. 421-434
[11] J. Goodwin and G. Chetty, "Blind Video Tamper Detection Based on Fusion of Source Features," in *Proc. International Conference on Digital Image Computing: Techniques and Applications* (DICTA 2011), Dec 6-8, (**2011**), pp. 608-613
[12] P. K. Atrey, W. Q. Yan, E. C. Chang, and M. S. Kankanhalli, "A Hierarchical SignatureScheme for Robust Video Authentication using Secret Sharing," in *Proc. 10th International Multimedia Modeling Conference* (MMM'04), Jan 5-7, (**2004**), pp. 330-337
[13] S. Upadhyay and S. K. Singh, "Video Authentication: Issues and Challenges. International Journal of Computer Science Issues, Vol. 9, Issue 1, No. 3, (**2012**), 409-418
[14] S. Upadhyay and S. K. Singh, "Learning Based Video Authentication using Statistical Local Information," in *Proc. International Conference on Image Information Processing* , Nov 3-5, (**2011**), pp. 1-6
[15] T. Shanableh, "Detection of Frame Deletion for Digital Video Forensics," Digital Investigation 10 (**2013**), pp. 350-360
[16] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double MPEG compression," in *Proc. 8th ACM workshop on Multimedia and security* (MM&Sec '06), Sept 26-27, (**2006**), pp. 37-47
[17] W. Wang and H. Farid, "Exposing digital forgeries in video by detecting double quantization," in *Proc. 11th ACM workshop on Multimedia and security* (MM&Sec '09), Sept 7-8, (**2009**), pp. 37-47
[18] M. K. Thakur, V. Saxena, and J. P. Gupta, "A Full Reference Algorithm for Dropped Frames Identification in Uncompressed Video Using Genetic Algorithm," International Journal of Digital Content Technology and its Applications, Vol. 6, No. 20, (**2012**), pp. 562-573
[19] G. Rascioni, S. Spinsante, and E. Gambi, "An Optimized Dynamic Scene Change Detection Algorithm for H.264/AVC Encoded Video Sequences," International Journal of Digital Multimedia Broadcasting, Vol. 2010, (**2010**), pp. 1-9
[20] W. S. Chau, O. C. Au, T. S. Chong, T. W. Chan, and C. S. Cheung, "Efficient Scene Change Detection in MPEG Compressed Video Using Composite Block Averaged Luminance Image Sequence," in *Proc. 5th International Conference on Information, Communications and Signal Processing*, (**2005**), pp. 688-691
[21] P. Seeling, "Scene Change Detection for Uncompressed Video," Technological Developments in Education and Automation, (**2010**), pp. 11-14
[22] http://compression.ru/video/quality_measure/metric_plugins/dfm_en.htm/. Accessed 30 Jan 2014
[23] 2013 Volvo FH Series on the Road, at www.youtube.com/watch?v=VZX-o9jzX0k. Accessed 30 Jan 2014
[24] The New Volvo FH, at www.youtube.com/watch?v=bQmmlIXS0fc. Accessed 30 Jan 2014
[25] Volvo Trucks–How Volvo FMX was tested, at http://www.youtube.com/watch?v=QokdT75uFf4, Accessed 30 Jan 2014
[26] All new Volvo FH16 750, at http://www.youtube.com/watch?v=XiK-qd8iNwY. Accessed 30 Jan 2014

## Authors

**Manish Kumar Thakur**. He received his M Tech in 2004 (from BIT Mesra) and Ph D in 2014 (from JIIT Noida) and currently working as Assistant Professor at JIIT Noida, India. He is author of around 15 research papers published in International Journals and Conferences. His research interests include video processing, data structures and algorithms.

**Vikas Saxena**. He did his Ph D in 2009 and currently working as Associate Professor at JIIT Noida, India. He has more than 35 publications in International journals and conferences. His expertise is in the field of Image processing, computer graphics and computer vision.

**J P Gupta**. Currently, he is Director Emeritus (QA) at Hydrocarbons Education and Research Society, New Delhi, India. He is an academician having more than 35 years experience including Professor at IIT Roorkee, Vice Chancellor at JIIT Noida, Galgotia University and Sharda University.