CHAPTER 5: THRESHOLD BASED NO REFERENCE SCHEME FOR FRAME DROP DETECTION

Frame drop or deletion is one of the common temporal tampering, where forger can drop or delete some frames in original video and create temporally tampered video. As discussed in Chapter 2, under no reference (NR) mode of tampering detection, it is mostly handled either using “Codec” based schemes or using classification based scheme.

We discussed in Chapter 2 that “Codec” based schemes are broadly meant for MPEG-x encoded videos, where tampering of frame drop shifts some $P$ frames from one GOP (group of pictures) to other, resulting into non-periodic increment of motion errors in the $P$ frames within each GOP, and hence the tampering of frame drop can be detected by tracing such non-periodic increments. As detailed in Chapter 2, these schemes are inefficient to identify exact location of tampering of frame drop, if intermediate frames in multiple GOPs are deleted, i.e. there are multiple sets of deleted frame where one set represents deletion of intermediate frames in one GOP.

Further, in previous chapter we have seen that the classification based schemes viz. scheme presented by Upadhyay and Singh in [50] as well as our proposed scheme with enhanced features are inefficient to detect the tampering of frame drop/deletion in such videos which were not involved in the training of the SVM (Support Vector Machines) classifier.

Therefore, to overcome these limitations, in this chapter we propose threshold based NR scheme to detect the tampering of frame drop/deletion in a video. The scheme is proposed for uncompressed videos, and hence capable to detect the tampering of frame drop in videos irrespective of the video encoding scheme.

Unlike GOPs (usually of smaller chunk of frame sequence) in compressed videos, the proposed scheme has been developed for the deletion of intermediate frames in scenes (usually a bigger chunk of video frames) in an uncompressed video. Recall our consideration about a scene; it is the sequence of successive frames having no abrupt change between any two frames in the sequence.

In the proposed scheme, we recommended three thresholds, viz. $DISP_{\text{similar}}$, $DISP_{\text{diff}}$, and $DISP_{\text{sig}}$, which help to decide the abrupt change or significant change between two
successive frames, besides the almost identical frames. Here, we first analyse the abrupt changes between two successive frames in a given video to split the video into different scenes. Further, these individual scenes have been analysed for frame deletion (if any) in the given video and hence, if there are multiple sets of dropped frames, where one set represents deletion of intermediate frames in one scene, may be detected. Conducted experiments shown significant improvement in the achieved accuracy over the schemes presented in Chapter 4.

In subsequent section, we discuss the thresholds and their recommended values besides a scheme to identify the abrupt change of scene and the scheme to detect the tampering of frame deletion. Section 5.1 presents the problem statement; frame drop detection scheme is proposed in Section 5.2; experimental analysis of the proposed scheme is presented in Section 5.3, followed by summary in Section 5.4.

5.1 Problem Statement

Let us consider an original video, $O$ with $m$ video frames, which is comprised of $k$ scenes. The video, $O$ is being tampered by dropping or deleting multiple sets of intermediate frames in video scenes, and thus creates a temporally tampered video, $T$ with $n$ video frames, where, $m \geq n$. Objectives are for given video, $T$, if, $O$ is not available (i.e. NR tampering detection):

(a) Examine the authenticity of given video, $T$ (i.e. identify $T$ as tampered or authentic video), and

(b) If, $T$ is identified as tampered video, then identify the exact location of tampering in the video, $T$, i.e. frame indices in $T$ after which frames were dropped or deleted.

The above problem is illustrated by an example in Figure 5.1, where original video $O$ ($m = 21$ frames) is comprised of 3 scenes ($k = 3$). The tampered video, $T$ ($n = 15$ frames) is created by deleting following two sets of intermediate frames in video scenes of the video, $O$

(a) Frames $O(3)$, $O(4)$, and $O(5)$ in scene, $S_1$ have been deleted

(b) Frames $O(17)$, $O(18)$, and $O(19)$ in scene, $S_3$ have been deleted

While verifying the authenticity of the “to be examined video”, $T$, it is expected to identify $T$ as tampered video and location of tampering of frame drop after frames, $T(2)$ and $T(13)$.

Next section presents the proposed scheme to verify the authenticity of the “to be examined video”, $T$ as well as the location of tampering of frame drop.
5.2 Proposed Scheme

As a solution of the problem defined in previous section, this section presents a tampering detection scheme where we have recommended various static and dynamic thresholds to verify the authenticity of a given video as well as the location of tampering of frame drop, if given video is found as tampered.

In this scheme, instead of tracing the tampering of frame drop in entire video, we split the video into different scenes and then each scene has been individually analysed for the frame deletion. Therefore, static thresholds have been proposed for two involved steps, viz. detection of change of scene (abrupt change) and detection of frame drop, whereas dynamic thresholds have been dynamically decided for individual video during the processing of involved steps.

Subsequently in this section we propose various thresholds, a scheme to detect the abrupt change of scene and the tampering detection scheme. Section 5.2.1 presents the thresholds; Section 5.2.2 presents the algorithm, \textit{detectSceneChange} to detect abrupt change of scene in a video; and Section 5.2.3 presents the scheme for detection of tampering of frame drop.

5.2.1 Thresholds

In previous chapter, we have trained the SVM classifier with four features; viz. Entropy, Average Object Area, Mean Squared Error (MSE), and Count of Displaced Blocks (DISP), where features MSE and DISP performed comparatively better to detect the tampering of
frame drop. Therefore, we have defined various thresholds involving the features, MSE and DISP and these thresholds have been used in the schemes presented in this chapter.

Usually videos have temporally redundant frames therefore the differences between two successive frames in a video are less until there is no abrupt change between two frames, i.e. abrupt change of scene. In Figure 5.2, we have shown successive frames in two videos, viz. “bus” and “tt” available at http://media.xiph.org/video/derf/, where first two pictures are the first two frames (i.e. frame 1 and frame 2) in the video “bus”, whereas last two pictures are the frame 56 and frame 57 in the video “tt”. Due to abrupt change, the difference between frame 56 and frame 57 in the video “tt” will be significantly large, whereas it will be less in the case of shown frames in video “bus”.

![Figure 5.2: Two successive frames in videos, “bus” and “tt”][93]

Considering that, there is a drop of 9 frames after the frame 2 in the video “bus”, we have shown in Figure 5.3, first four frames in the tampered video, “busT” created from video, “bus”. These first four frames are actually the frame 1, frame 2, frame 12, and frame 13 in the original video, “bus”. With reference to the feature DISP (recall, it gives the count of $8 \times 8$ blocks in a frame not present in another frame) following inferences have been made from the successive frames of original and tampered videos have shown in Figure 5.2 and Figure 5.3.

(a) Due to gradually changing frames, there may be least count of displaced blocks, DISP between frame 1 and frame 2 in video “bus”, i.e. except few, remaining $8 \times 8$ blocks in frame 1 may found present in frame 2.

![Figure 5.3: First four frames in the tampered video “busT”][93]
(b) Due to abrupt change between frame 56 and frame 57 in the video “tt” the count of displaced blocks, DISP will be significantly large, \( i.e. \) except few, remaining \( 8 \times 8 \) blocks in frame 56 may not found in frame 57.

c) Due to deletion of 9 frames after frame 2 in the video “bus”, the count of displaced blocks, DISP between frame 2 and frame 3 in the tampered video, “busT”, which are actually frame 2 and frame 12 in original video, “bus” will be significant \( i.e. \) significantly more than the scenario presented in (a) and significantly less than the scenario presented in (b)). Here, there may be the possibility of presence of many \( 8 \times 8 \) blocks in these successive frames in the tampered video of “bus” as well as many \( 8 \times 8 \) blocks may not present in these successive frames.

Based on above discussion, we define following thresholds which involve the feature DISP, where DISP between two frames, \( \text{viz.} F_i \) and \( F_j \) has been computed using the algorithm \( \text{blkDisp} \) presented in chapter 4.

\[
DISP_{diff} = \frac{\text{blkDisp}(F_i, F_j)}{\text{Count of } 8 \times 8 \text{ blocks in the given video frames}}
\]

where, \( F_i \) and \( F_j \) are entirely different frames \( \text{viz.} \) frame 56 and frame 57 in video, “tt”, as shown in Figure 5.2) and the algorithm \( \text{blkDisp} \) returns the count of displaced \( 8 \times 8 \) blocks.

We have conducted experiments with frames in cif videos (frame size of \( 352 \times 288 \) pixels, \( i.e. \) there are 1584 blocks of size \( 8 \times 8 \) pixels in each frame) and explored such \( F_i \) and \( F_j \) where both frames are entirely different. Based on these experiments involving entirely different frames, we recommend its value as 0.79.

\[
DISP_{similar} = \frac{\text{blkDisp}(F_i, F_j)}{\text{Count of } 8 \times 8 \text{ blocks in the given video frames}}
\]

Based on these experiments involving entirely different frames, we recommend its value as 0.79.
where, $F_i$ and $F_j$ are either identical or gradually changing frames (viz. frame 1 and frame 2 in video, “bus”, shown in Figure 5.2) and the algorithm $blkDisp$ returns the count of displaced $8 \times 8$ blocks.

To recommend the value for the threshold, $DISP_{similar}$, we have conducted experiments with frames in cif videos (i.e. there are 1584 blocks of size $8 \times 8$ in each frame) and investigated such $F_i$ and $F_j$ where both frames are either identical or gradually changing frames. Based on the experiments with such frames in cif videos, we recommend its value as 0.16.

$DISP_{sig}$: This is the last threshold defined with feature, $DISP$ and represents the ratio between count of $8 \times 8$ blocks of frame, $F_i$ not present in frame $F_j$, if there is significant change between both frames (which may be due to frame drop) and count of $8 \times 8$ blocks in the given video frames.

$$DISP_{sig} = \frac{blkDisp(F_i,F_j)}{Count \ of \ 8 \times 8 \ blocks \ in \ the \ given \ video \ frames}$$

where, $F_i$ and $F_j$ are neither gradually changing frames nor entirely different frames (viz. frame 2 and frame 3 in the tampered video, “busT”, shown in Figure 5.3) and the algorithm $blkDisp$ returns the count of displaced $8 \times 8$ blocks.

We have randomly dropped 5 to 20 successive frames after a frame in a video to get such $F_i$ and $F_j$ (viz. if there are drop of 10 frames, then $F_j$ is actually the $(i+10)$th frame in that video), where we ensured that these $F_i$ and $F_j$ are the frames in a single video scene. Based on the experiments with such $F_i$ and $F_j$, we recommend its value as 0.52. These experiments have been conducted with frames in cif videos (i.e. there are 1584 blocks of size $8 \times 8$ pixels in each frame).

In addition to above thresholds, we have defined one more threshold, viz. $MTH_{Low}$, which has been used in the tampering detection schemes presented in chapter 6 and chapter 7 in this thesis.

$MTH_{Low}$: It represents the maximum Mean Squared Error (MSE) between two video frames ($F_i$ and $F_j$) such that we can call both frames nearly same (i.e. in between identical frames and gradually changing frames). Based on conducted experiments with such frames, $F_i$ and $F_j$, in cif videos, we recommend its value as 524.
Next section presents the algorithm, \textit{detectSceneChange}, which has been used to identify the abrupt change of scene between two successive frames in a video.

5.2.2 Abrupt Scene Change Detection

Abrupt scene change detection and accordingly compute the count of scenes in a video is one of the pre-processing steps used in the tampering detection scheme proposed in next section as well as the schemes proposed in chapter 6 and chapter 7 in this thesis.

In this section, we propose an algorithm, \textit{detectSceneChange} to detect the abrupt scene change, where we used the feature, MSE to compare adjacent frames, $F_i$ and $F_{i+1}$ for $\forall i = 1$ to $m - 1$, where $m$ is the count of frames in the given video. These MSEs have been used to dynamically compute the thresholds, $MSE_{high}$ and $MSE_{global}$ in a given video as follows:

(a) The dynamic threshold, $MSE_{global}$ represents the average of all MSEs between each successive frames in a given video and computed as

$$MSE_{global} = \frac{\sum_{i=1}^{m-1} D(i)}{m - 1}$$

where, $D(i)$ stores the MSE between $i^{th}$ and $(i + 1)^{th}$ frames in the given video and $m$ is the count of frames in the given video.

(b) The dynamic threshold, $MSE_{high}$ represents the average of such MSEs which are greater than equal to $MSE_{global}$

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

where, $D(i)$ stores the MSE between $i^{th}$ and $(i + 1)^{th}$ frames in the given video; $m$ is the count of frames in the given video; and $t$ is the count of such MSEs between $i^{th}$ and $(i + 1)^{th}$ frames which are greater than or equal to $MSE_{global}$.

The algorithm \textit{detectSceneChange} is presented in Figure 5.4. Here, based on the computed dynamic thresholds in the given video, we have rejected such successive frames for the possibility of abrupt scene change, if the MSE between these successive frames is less than $m_1 \times MSE_{high}$, where $m_1 < 1$ is a margin parameter and need to be tuned. Further, we have included all such successive frames as the possibility of abrupt scene change, if the MSE
between these successive frames is greater than \( m_2 \times \text{MSE}_{\text{high}} \), where \( m_2 > 1 \) is a margin parameter and need to be tuned.

**Algorithm: detectSceneChange(V)**  
{Detects the abrupt change of scene in video, V}

\[
\begin{aligned}
m & \quad \text{(Count of frames in video, V)} \\
count & \quad \text{(A variable which stores the count of scenes in video, V)} \\
D[1 \ldots m-1] & \quad \text{(A vector which stores the MSE between two adjacent frames in V)} \\
mark[1 \ldots m-1] & \quad \text{(A vector which stores Boolean value)} \\
blkCount & \quad \text{(It is the count of 8 \times 8 blocks in a frame of video, V)} \\
\end{aligned}
\]

\[
\begin{aligned}
\text{begin} & \\
\text{initialize} & \quad \text{count} \leftarrow 0, i \leftarrow 1, \text{temp} \leftarrow 0, t \leftarrow 0, \text{var} \leftarrow 0 \\
\text{for} & \quad i \leftarrow 1 \text{ to } m-1 \text{ do} \\
\text{D}[i] & \quad = \text{MSE}(V(i), V(i+1)) \quad \{ \text{MSE returns the MSE between } i^{th} \text{ and } i+1^{th} \text{ frames in V} \} \\
\text{var} & \quad \leftarrow \text{var} + D[i] \\
\text{end for} \\
\text{MSE}_{\text{global}} & \quad = \text{var}/(m-1) \\
\text{LABEL 1} & \quad \{ \text{A label to repeat the steps} \} \\
\text{var} & \quad \leftarrow 0 \\
\text{for} & \quad i \leftarrow 1 \text{ to } m-1 \text{ do} \\
\text{if} & \quad \text{D}[i] \geq \text{MSE}_{\text{global}} \text{ then} \\
\text{var} & \quad \leftarrow \text{var} + D[i] \\
\text{t} & \quad \leftarrow t + 1 \\
\text{end if} \\
\text{end for} \\
\text{MSE}_{\text{high}} & \quad = \text{var}/t \\
\text{for} & \quad i \leftarrow 1 \text{ to } m-1 \text{ do} \\
\text{mark}[i] & \quad \leftarrow 0 \\
\text{end for} \\
\text{temp} & \quad \leftarrow 0 \\
\text{for} & \quad i \leftarrow 1 \text{ to } m-1 \text{ do} \\
\text{p}_1 & \quad \leftarrow m_1 \times \text{MSE}_{\text{high}} \\
\text{p}_2 & \quad \leftarrow m_2 \times \text{MSE}_{\text{high}} \quad \{ m_1 < 1 \text{ and } m_2 > 1 \text{ are the margins and need to be tuned} \} \\
\text{if} & \quad \text{D}[i] \geq \text{p}_1 \text{ and } \text{D}[i] < \text{MSE}_{\text{high}} \text{ then} \\
\text{if} & \quad \text{blkDisp}(V(i), V(i+1))/\text{blkCount} \geq \text{DISP}_\text{diff} \text{ then} \\
\text{mark}[i] & \quad \leftarrow 1 \\
\text{count} & \quad \leftarrow \text{count} + 1 \\
\text{temp} & \quad \leftarrow \text{temp} + 1 \\
\text{end if} \\
\text{end if} \\
\text{if} & \quad \text{D}[i] \leq \text{p}_2 \text{ and } \text{D}[i] > \text{MSE}_{\text{high}} \text{ then} \\
\text{if} & \quad \text{blkDisp}(V(i), V(i+1))/\text{blkCount} \leq \text{DISP}_\text{sig} \text{ then} \\
\text{continue the loop} \\
\text{else} \\
\text{mark}[i] & \quad \leftarrow 1 \\
\text{count} & \quad \leftarrow \text{count} + 1 \\
\text{temp} & \quad \leftarrow \text{temp} + 1 \\
\text{end if} \\
\end{aligned}
\]
Further, we have explored such successive frames for the count of displaced blocks (using scheme \( \text{blkDisp} \) presented in chapter 4), if the MSE between these successive frames in the given video is at the margin of \( \text{MSE}_{\text{high}} \), i.e. between \( m_1 \times \text{MSE}_{\text{high}} \) and \( m_2 \times \text{MSE}_{\text{high}} \).

Here, we have included such successive frames as the possibility of abrupt scene change, where the ratio between count of displaced \( 8 \times 8 \) blocks between these frames and count of total blocks in a video frame is greater than equal to \( \text{DISP}_{\text{diff}} \). If, any such successive frames have been found, then we increased the value of a variable \( \text{temp} \) by 1, where \( \text{temp} \) was initially 0. Further, we have rejected such successive frames as the possibility of abrupt scene change, where the ratio between count of displaced \( 8 \times 8 \) blocks between these frames and count of total blocks in a video frame is less than equal to \( \text{DISP}_{\text{sig}} \).

At this stage, if \( \text{temp} \geq 1 \), then we have reported all the successive frames which have been identified as the possibility of abrupt scene change as the successive frames having abrupt scene change. Otherwise, we have repeated the process by making \( \text{MSE}_{\text{global}} \) as \( \text{MSE}_{\text{high}} \) and recomputed the \( \text{MSE}_{\text{high}} \) for the new \( \text{MSE}_{\text{global}} \). We have repeated these steps until \( \text{temp} \geq 1 \) or there is no such successive frames left in the video, where the MSE between those frames are greater than \( \text{MSE}_{\text{global}} \). Hence, the thresholds, \( \text{MSE}_{\text{global}} \) and \( \text{MSE}_{\text{high}} \) keep on dynamically changing depending on the actual count of abrupt changes in a given video.
there is no such abrupt change of scene in the given video, then the thresholds $MSE_{global}$ as $MSE_{high}$ will keep on changing till the described condition.

Next section presents the proposed scheme for the detection of frame drop or deletion.

5.2.3 Detection of Frame Drop

As discussed earlier, in the proposed tampering detection scheme, we first split the video into different scenes and each scene has been independently examined for the frame drop. Here, the dynamic thresholds, viz. $MSE_{high}$ and $MSE_{global}$ are applied for individual scenes and accordingly two successive frames have been checked for the count of displaced blocks, if the difference between these successive frames are greater than the threshold $MSE_{high}$ for that scene. We have reported the first frame of such successive frames as the frame after which there is tampering of frame deletion, if the count of displaced blocks between these successive frames is greater than the static threshold, $DISP_{sig}$.

Subsequently, we present the detailed steps for detection of the tampering of frame drop.

**Step 5.2.3.1:** Input the “to be examined video”, $T$ having $n$ frames as $T(1)$, $T(2)$, ... $T(n)$.

In the example of Figure 5.1, there are 15 frames ($n = 15$) in the video, $T$, viz. $T(1)$ to $T(15)$.

**Step 5.2.3.2:** Apply the abrupt scene change detection scheme, $detectSceneChange$ to identify different scenes in the video, $T$ and store frame indices of each scene of $T$ into individual bins (say $k$ bins have been formed).

In the preceding example, the scheme, $detectSceneChange$ may successfully identify all the 3 scenes in the video, $T$. Frame indices of each frame in each scene have been stored into different bins. Figure 5.5 presents such bins (or scenes) for the video, $T$ of Figure 5.1.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Frame#1</td>
<td>Frame#5</td>
<td>Frame#12</td>
</tr>
<tr>
<td><em>i.e. T(1) or O$_{1}$</em></td>
<td><em>i.e. T(5) or O$_{9}$</em></td>
<td><em>i.e. T(5) or O$_{15}$</em></td>
</tr>
<tr>
<td>Frame#2</td>
<td>Frame#6</td>
<td>Frame#13</td>
</tr>
<tr>
<td><em>i.e. T(2) or O$_{2}$</em></td>
<td><em>i.e. T(6) or O$_{9}$</em></td>
<td><em>i.e. T(13) or O$_{16}$</em></td>
</tr>
<tr>
<td>⋯</td>
<td>⋯</td>
<td>⋯</td>
</tr>
<tr>
<td>Frame#4</td>
<td>Frame#11</td>
<td>Frame#15</td>
</tr>
<tr>
<td><em>i.e. T(1) or O$_{7}$</em></td>
<td><em>i.e. T(11) or O$_{14}$</em></td>
<td><em>i.e. T(15) or O$_{21}$</em></td>
</tr>
</tbody>
</table>

**Figure 5.5:** Bins containing frame indices of each frame in each scene in $T$
Step 5.2.3.3: For ∀ \( i = 1 \) to \( k \), where \( k \) is the count of scenes or bins, compare the adjacent frames, \( i.e. \) the \( j^{th} \) and \( (j+1)^{th} \) frames in each scene, \( S_i \) (frame indices are available in respective bins) using MSE and store these differences into matrix, \( D(i,j) \).

\[
D(i,j) = \text{MSE} \left( T(B(i,j)), T(B(i,j+1)) \right) \quad \text{for } \forall i = 1 \text{ to } k \text{ and } \forall j = 1 \text{ to } f \text{count}(B(i))
\]

Where, \( B(i,j) \) is the index of \( j^{th} \) frame in scene \( S_i \) in video, \( T; T(i) \) is the \( i^{th} \) frame in video, \( T; f \text{count}(B(i)) \) returns count of frames in scene \( S_i \); and \( k \) is the count of scenes or bins.

Step 5.2.3.4: For each bin, \( B(i) \), where \( i = 1 \) to \( k \), do following

(a) Compute average of MSE, \( AMSE \), \( i.e. \) \( MSE_{global} \) for bin \( B(i) \) as follows

\[
AMSE(i) = \frac{\sum_{j=1}^{f \text{count}(B(i))} D(i,j)}{f \text{count}(B(i))}
\]

(b) Compute average of high MSE, \( HMSE \), \( i.e. \) \( MSE_{high} \) for bin, \( B(i) \) as follows

\[
HMSE(i) = \frac{\sum_{j=1}^{f \text{count}(B(i))} D(i,j)}{t} \quad \text{if } D(i,j) \geq AMSE(i) \text{ and } t \text{ is count of such } D(i,j)
\]

(c) Use the subroutine, \( blkDisp \) (presented in chapter 4) to compute count of displaced block \( BD(i,j) \) between frames in video, \( T \) indexed at \( B(i,j) \) and \( B(i,j+1) \), if \( D(i,j) \geq HMSE(i) \).

\[
BD(i,j) = blkDisp \left( T(B(i,j)), T(B(i,j+1)) \right) \quad \text{if } D(i,j) \geq HMSE(i)
\]

(d) Report the tampering of frame drop after frame index \( B(i,j) \), if

\[
BD(i,j)/bCount \geq DISP_{sig}, \quad \text{ // } bCount \text{ is the count of } 8 \times 8 \text{ blocks in a frame}
\]

Step 5.2.3.5: Report \( T \) as tampered video, if there is reporting of tampering in Step 5.2.3.4 (d).

In the preceding example, there might be possibility that \( D(1,2) \) \( i.e. T(2) \), which is actually frame, \( O_2 \) in original video, \( O \) is greater than high MSE of scene or bin \( B(1) \) and \( BD(1,2) \geq DISP_{sig} \). Thus report \( B(1,2) \) \( i.e. T(2) \) as frame indices in video, \( T \) after which there is a tampering of frame drop. Similarly other frame indices can also be obtained as frame indices in video, \( T \) after which there is a tampering of frame drop.
5.3 Experimental Analysis

There are majorly two involved steps in the tampering detection scheme presented in previous section, viz. identification of scenes (abrupt change) in the given video and for each scene identification of dropped frames (if any).

Based on the conducted experiments, this section presents the analysis of the performance of proposed tampering detection scheme which incorporates the algorithm detectSceneChange. Section 5.3.1 presents the video data sets, whereas Section 5.3.2 presents the performance analysis.

5.3.1 Video Data Sets

This section presents the sets of original and tampered videos which have been used to conduct the experiments. We have used 4 original videos (presented in Figure 5.6) to create tampered videos. These original videos include

V_1: The video, “2013 Volvo FH Series on the Road”, available at www.youtube.com/watch?v=VZX-o9jzX0k.

V_2: The video, “Volvo Trucks – How Volvo FMX was tested”, available at www.youtube.com/watch?v=QokdT75uFf4.


V_4: The video “All new Volvo FH16 750”, available at www.youtube.com/watch?v=XiK-qd8jNwY.

![Figure 5.6: Set of videos used as original videos][1]

These original videos are available in compressed form; therefore, before getting these videos tampered, we uncompressed these videos and for uniformity in the frame size of each video, scaled to 352 x 288 pixels (i.e. cif video). We manually counted the number of scenes (i.e.}
the abrupt change of scene between two successive frames) and the count of frames in each scene in the uncompressed original videos.

Further, we have created 15 copies of each original video and manipulated them by randomly dropping intermediate frames (5 to 15 frames) in some scenes, i.e. copies of each original video were tampered with multiple sets of dropped frames. In total, we have created 15 tampered videos (T₁ to T₁₅) for each original video, Vᵢ. Altogether we have created 60 tampered videos (15 tampered videos × 4 original videos). It is important to mention here that deletion of intermediate frames in some scenes in the copy of original video will not change the count of scenes in tampered videos, i.e. it will remain same as in the original video.

Table 5.1 details the count of scenes in original videos as well as the actual count of sets of dropped frames (where one set of dropped frames represent the deletion of 5 to 15 intermediate frames in a video scene in the copy of original video) in each tampered videos, i.e. T₁, T₂, . . . T₁₅ created from the copies of original videos, Vᵢ.

**Table 5.1: Count of scenes in original videos and actual count of sets of dropped frames in tampered videos**

<table>
<thead>
<tr>
<th>Original Video</th>
<th>Count of Scenes</th>
<th>Actual count of sets of dropped frames in each tampered videos (viz. T₁ to T₁₅) created from video copies of respective original videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>V₁</td>
<td>28</td>
<td>T₁</td>
</tr>
<tr>
<td></td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>V₂</td>
<td>62</td>
<td>15</td>
</tr>
<tr>
<td>V₃</td>
<td>49</td>
<td>12</td>
</tr>
<tr>
<td>V₄</td>
<td>32</td>
<td>12</td>
</tr>
</tbody>
</table>

As seen in Table 5.1, we have dropped 10 sets of intermediate frames from 10 scenes in the copy of original video, V₁ to create the tampered video, T₁. Similarly we have dropped different sets of intermediate frames to create other tampered videos from the copies of video, Vᵢ. In total we have dropped 198 sets of intermediate frames from 198 scenes in the copies of video, Vᵢ to create 15 tampered videos (T₁ to T₁₅). This count is 330, 279, and 210 for the copies of original videos, V₂, V₃, and V₄ respectively. Altogether, we have dropped 1017 sets of intermediate frames from 1017 scenes from the copies of respective original videos to create 60 tampered videos.

In subsequent section, we present the analysis of the performance of the proposed scheme.
5.3.2 Performance Analysis

We have conducted experiments with the help of 60 tampered videos created from 4 original videos to analyse the performance of the proposed scheme, where the objectives are to verify the authenticity of the “to be examined videos” and identify the location of tampering of frame drop, if video is found as tampered.

While conducting the experiments, we have observed the performance of the involved steps in the proposed scheme, viz. identification of scenes in the “to be examined videos” and identification of the tampering location and accordingly videos to be categorised either as authentic or tampered video.

Figure 5.7 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 video, $V_1$. Based on the observations, Figure 5.7 (a) presents the plot between

(i) Actual count of scenes in each tampered video created from original video, $V_1$.
(ii) Count of correctly detected scenes in each tampered video. – i.e. True Positive.
(iii) Count of incorrectly detected scenes (i.e. manually we observed no abrupt change of scene between two frames, but the scheme incorrectly identified the abrupt change and accordingly identified single scene as two scenes) in each tampered video, viz. $T_1$ to $T_{15}$ created from original video, $V_1$. – i.e. False Positive.
(iv) Count of incorrectly rejected scenes in each tampered video viz. $T_1$ to $T_{15}$ created from original video, $V_1$. – i.e. False Negative.

whereas, Figure 5.7 (b) presents the plot between

(i) Count of sets of dropped frames which have been actually dropped from some scenes in which intermediate frames have been dropped from the copies of original video, $V_1$.
(ii) Count of correctly detected sets of dropped frames, i.e. the count of correctly identified frame indices after which there is the tampering of frame drop in each tampered video (viz. $T_1$ to $T_{15}$) created from original video $V_1$. – i.e. True Positive.
(iii) Count of incorrectly detected sets of dropped frames in each tampered video (viz. $T_1$ to $T_{15}$) created from original video $V_1$. – i.e. False Positive.
(iv) Count of incorrectly rejected sets of dropped frames in each tampered video (viz. $T_1$ to $T_{15}$) created from original video $V_1$. – i.e. False Negative.
Figure 5.7: Performance of the frame drop detection scheme involving tampered videos created from video, $V_1$

Similar plots have been presented for the tampered videos (viz. $T_1$ to $T_{15}$) created from $V_2$, $V_3$, and $V_4$ in Figure 5.8, Figure 5.9, and Figure 5.10 respectively.

Figure 5.8: Performance of the frame drop detection scheme involving tampered videos created from video, $V_2$
Out of 2565 scenes (abrupt change) in all 60 tampered videos (i.e. $(28 + 62 + 49 + 32) \times 15$), the algorithm, `detectSceneChange` identified 2243 scenes, i.e. achieved percentage accuracy to identify the scenes in temporally tampered videos (where tampering is due to frame drop) is around 87.5%. Further, based on the observations while conducting the experiments
(plotted in Figure 5.7 to Figure 5.10), performance of the tampering detection scheme has been analysed in terms of percentage accuracy achieved to correctly identify the location of tampering of frame drop as well as categorisation of the “to be examined video” either as tampered video or authentic video, as follows

(a) Percentage accuracy to correctly identify the location of tampering (i.e. sets of frame drop) has been computed as follows

\[
\text{Accuracy (in \%)} = \frac{\text{Count of correctly identified sets of dropped frames in all the tampered videos}}{\text{Total count of set of dropped frames while creation of tampered videos}} \times 100
\]

As discussed in Section 5.3.1, total count of sets of dropped frames is 1017, whereas count of correctly identified sets of dropped frames is 866. Table 5.2 presents the actual and detected counts of sets of dropped frames and the achieved accuracy (in %).

<table>
<thead>
<tr>
<th>Original Video (from which tampered videos were created)</th>
<th>Count of sets of dropped frames, i.e. tampering of frame drop (in all tampered videos)</th>
<th>Accuracy (in %)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Detected</td>
</tr>
<tr>
<td>V₁</td>
<td>198</td>
<td>175</td>
</tr>
<tr>
<td>V₂</td>
<td>330</td>
<td>272</td>
</tr>
<tr>
<td>V₃</td>
<td>279</td>
<td>235</td>
</tr>
<tr>
<td>V₄</td>
<td>210</td>
<td>184</td>
</tr>
</tbody>
</table>

As seen from Table 5.2, the achieved average accuracy to detect the location of tampering of frame drop is between 82.42% and 88.38%.

(b) Percentage accuracy to correctly identify the “to be examined video” either as tampered or authentic has been computed as follows

\[
\text{Accuracy (in \%)} = \frac{\text{Count of videos identified as tampered video}}{\text{Total count of tampered videos}} \times 100
\]

We have conducted the experiments with 60 tampered videos (15 each for 4 original videos). As seen in Figure 5.7 to Figure 5.10, at least one set of dropped frames has been successfully detected in each of the “to be examined videos”, and hence identified as
tampered video. Hence, the achieved accuracy to identify the “to be examined videos” either as authentic or tampered videos is 100%.

Next section presents the summary of the Chapter 5.

5.4 Summary

In this chapter, we have proposed a no reference (NR) scheme to detect the tampering of frame drop. The scheme has been proposed to overcome the limitations with “Codec” based schemes as well as to improve the performance to detect the tampering of frame drop which is not satisfactory with the classification based schemes presented in Chapter 4.

The tampering detection scheme proposed in this chapter successfully categorises the “to be examined video” as tampered video as well as identifies the location of tampering (i.e. location of set of dropped frames). The scheme has been tested on sixty tampered videos (where videos were tampered with multiple instances or sets of dropped frames) created from copies of four original videos. The achieved average accuracy to identify the location of tampering is between 82.42% and 88.38%, whereas it is 100% to categorise the “to be examined video” as tampered video. With the same sets of tampered videos, used in this chapter, we re-examine the performance of the classification based schemes presented by Upadhyay and Singh in [50] as well as our proposed scheme with enhanced features (presented in Chapter 4). Like the accuracies reported in Chapter 4 for unknown videos, while re-examination, we have found the achieved average accuracies to identify the location of tampering as around 30% and 35% respectively for the feature subsets, Set 3 (features as Entropy and Average Object Area) and Set 4 (features as MSE and DISP).

Hence, the scheme proposed in this chapter has shown the significant improvement over the classification based schemes presented in Chapter 4. Further, the scheme proposed in this chapter is capable to identify the location of tampering of frame drop when there are multiple sets of dropped frames, thus overcomes the limitations of “Codec” based schemes, such as the scheme presented by Wang and Farid in [75].