Journal of Theoretical and Applied Information Technology
31st July 2018. Vol.96. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

DYNAMIC DUAL THRESHOLD BASED SCHEMES FOR ABRUPT SCENE CHANGE DETECTION IN VIDEOS

MANISH K THAKUR
Department of CSE & IT, Jaypee Institute of Information Technology, India
E-mail: mthakur.jiit@gmail.com

ABSTRACT

Detection of abrupt and gradual change of scene in a video is an active area of research for last many years due to its applicability in many applications, *viz*. content based image retrieval, computer vision, *etc*. This paper presents dynamic dual threshold based schemes to detect the abrupt scene change in a video. Proposed schemes use static as well as dynamic thresholds defined over set of features proposed in this paper. These features include Mean Squared Error (MSE), Entropy, and Count of displaced blocks. Extensive experiments have been conducted to recommend the values for the proposed thresholds to efficiently detect the abrupt scene change in videos. Experimental results show the good accuracy to detect the abrupt change of scene.

Keywords: *Abrupt Change of Scene Detection, Entropy, Mean Squared Error, Histogram, Count of Displaced Blocks*

1. INTRODUCTION

For years, visual information (images and videos) is one of the most effective modes of representing and communicating information among end users. Visual information in form of continuous stream (*i.e.* video sequence or video) represents the broader spectrum of information [1-2]. In current era, visual information available at different social networking websites *viz*. YouTube, facebook, and Instagram, *etc*. are effectively used to fulfill different objectives, *viz*. knowledge acquisition, advertisement, *etc*. whereas, video conferencing and video chatting (using Skype, *etc*.) are often used for communicating information among end users.

A video sequence (or video) consists of time ordered sequence of video frames, where each video frame represents orthogonal bitmap images of $M \times N$ pixels. Successive video frames represent the temporal variation and primary reason for motion or moving objects in a video [3-4]. Series of continuous frames make a scene and set of scenes resulted into a video [5], *i.e.* a video is a sequence of continuous or successive scene (s). There is change of context or information between two successive scenes in a video.

Due to wide applicability, detection of change of scene in a given video is always an attractive domain for researchers for last many years. Splitting the video into different scenes (or video segmentation) is one of the primary steps in video indexing, video tagging, scene based watermarking, content based video retrieval, and computer vision, *etc* [5]. Accurately and efficiently identified scenes in videos are the basic building block in the listed application domains.

Identification of different scenes in video footages captured by CCTV based surveillance system is often required to analyze the crimes or security loopholes in the concerned area or locality. Often the identified scenes in videos are useful in detection of the forgeries with videos. Figure 1 presents a clip of the movie Speed (released in 1994) where the doctoring with the surveillance video to conceal the activities on the bus is detected by the movie character after seeing the repeated scenes [6]. In the area of computer vision, different machines (*viz*. Support Vector Machine, Neural Network, *etc*.) are used to train the videos. For training purpose, splitting of videos into frames and scenes are often required.

In above applications, volume of the video contents as well as the amount of information contained in each video are day by day increasing and ultimately raising challenges for research community to reliably and efficiently retrieve or search relevant information from a large collection of videos. The volume of required data processing and reliability of the scheme to detect the change of scene in videos always motivate researchers to come with efficient and reliable scheme.

Journal of Theoretical and Applied Information Technology
31st July 2018. Vol.96. No 14
© 2005 – ongoing JATIT & LLS
ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Figure 1: An example where repeated scenes have been detected in the surveillance video clip doctored by the characters in the movie Speed [6]

In literature, depending upon amount of change between the contents of successive frames in a video, change of scene between two consecutive video frames is majorly defined in following ways: (1) gradual change between successive frames of videos and (2) abrupt change between successive frames of videos [2-3]. For years, abrupt scene change detection have been used as pre or post processing steps in many applications *viz*. video compression, video indexing, video database navigation, video summarization, video editing, and video forgery detection, *etc* [7-13].

In recent years, several contributions have been made by researchers to efficiently detect the abrupt scene change in a video by identifying the similarity or difference between adjacent frames using various metrics, *viz*. histogram, mean squared error, and template matching, *etc*. These techniques have been often used to compute the inter-frame difference in videos and effectively applied to find the abrupt change of scene. Subsequently we discuss some of the contributions to detect the abrupt scene change in videos.

In [7], Yu and Srinath proposed a threshold based scheme to detect the abrupt scene change. With the help of color histogram of successive frames, authors claimed to achieve good accuracy. In another contemporary work, similar approach have been used by the researchers at Moscow State University (MSU), where plug-in to detect the abrupt scene change have been developed using block based color histogram [14].

Authors in [10] proposed adaptive threshold based scheme to detect the scene change in videos, whereas in [9], authors presented the research directions and set of criteria for measuring and comparing the performances of various schemes of scene change detection.

Based on the correlation between histogram of first frame in a video with histogram of other frames of the video under consideration, authors in [15] proposed a scheme to detect the abrupt change of scene and gradual change of scene in that video. Based on the sharp changes in the correlation values of the histograms of the first frame and other frames, abrupt scene changes in a video were claimed to be effectively detected.

In another contemporary work, authors in [5] proposed a two-pass algorithm to detect the abrupt change of scene in videos. In this work, authors used color histogram based frame differences and proposed a sliding window detector based scene change detection scheme.

In all, the basis of these discussed schemes is thresholds (defined over different metrics) to differentiate the contents of successive frames in a video for consideration of abrupt change of scene. These thresholds are either static (*i.e.* having some pre-recommended values for all videos) or dynamic (*i.e.* computed and decided for individual videos) and defined over various features or metrics, includes MSE, Histogram, Entropy [16], *etc*. Performance (how accurately abrupt scene change has been detected) of the threshold based abrupt scene change detection scheme depends on the kind of features (or set of features) used as thresholds and their recommended values.

In this paper, we present threshold based schemes to detect the abrupt scene change, where, we defined static thresholds, recommended their values after extensive experiments along with dynamic dual thresholds which are dynamically computed for individual videos.

Apart from introduction, remaining paper is organized as follows: Section 2 presents the prerequisites in terms of features, static thresholds and their recommended values, and a scheme which is used as a pre-processing step for the scene change detection schemes presented in Section 3; experimental details, observed performance, and comparative analysis have been presented in Section 4 followed by conclusion and references.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2. THRESHOLDS AND PRE-PROCESSING

This section presents the features and the static thresholds used in this paper along with a scheme to compute the count of displaced blocks between two successive video frames. This scheme (count of displaced blocks) is used as a pre-processing step for the scene change detection scheme presented in next section.

As discussed in previous section, many features have been used by researchers to compare adjacent frames in a video for similarity or dissimilarity of the contents. Absolute difference and MSE are two of such features, where pixel by pixel absolute error (for absolute difference) or pixel by pixel squared error (for MSE) between two successive frames (F_C and F_N) is the basis of comparison. MSE is the mean of the squares of the pixel by pixel errors between two frames (say C and N) and computed as follows (Equation 1):

$$MSE(C,N) = \frac{1}{A\times B}\sum_{i=1}^{A}\sum_{j=1}^{B}[C(i,j) - N(i,j)]^2 \quad (1)$$

where, A and B (in $A \times B$ pixels) are respectively height and width *i.e.* size of the video frames C and N.

Lesser value of MSE between two video frames (C and N) represents the least difference between them, *i.e.* both frames (C and N) are almost identical. Larger values of MSE between two video frames (C and N) represents the significant difference between them to be considered as ending and starting frames of two different scenes. However, it is misleading while abrupt scene change detection, as there might not be scene change, even though the error between successive frames may be significantly large due to shifting of blocks (8×8 or 16×16).

Hence, in this paper we defined a new feature and named it count of displaced blocks (DISP). It represents the count of 8×8 blocks in a frame which are not present in successive frame. This feature is recently proposed by the author (s) in [16] and inspired by the feature, block displacement, mostly used in block based video compressions [3-4] where, block size is of either 8×8 or 16×16.

Figure 2 presents a scheme, $blkDisp(\,)$ recently proposed by the author (s) in [16] and computes the count of 8×8 blocks in a frame (F_C) not present in another frame (F_N) *i.e.* it is used to compute the feature DISP between two frames (F_C and F_N). The scheme, $blkDisp(\,)$ is built around following two features, (a) Entropy, which is the statistical measure of the randomness, and (b) MSE.

In the scheme, $blkDisp(\,)$, Entropy of each 8×8 block in the input frame F_C have been computed and scaled with two margin parameters, m_1 and m_2 to accommodate the boundary (little bit upper and little bit lower) values. Further, the computed range (considering the boundary cases too) of the considered block of the input frame F_C is used to find corresponding probable matching block (8×8) or blocks in another input frame F_N. Later, the probable matching has been considered as the perfect matching block (8×8) of F_C present in F_N, if the MSE of these blocks (8×8) are in considerable limit. The considerable limit is decided using block based MSE threshold, $bThr$. In [16] we recommended its ($bThr$) value as 60, but after conducting further extensive experiments over various videos of different nature having several instances of (1) abrupt change of scene, (2) gradual change of scene, and (3) no change of scene, we recommend its ($bThr$) value as 62.

Utilizing the feature, DISP, we defined two static thresholds in [16] and present here as follows along with their recommended values:

$DISP_{gradual}$: It is one of the static thresholds defined in this paper. Considering gradually changing successive frames, say F_i and F_{i+1} (*i.e.* no abrupt change between these frames), this threshold is defined as the ratio between count of 8×8 blocks of a video frame, F_i not present in adjacent frame F_{i+1} and count of 8×8 blocks in given video frames.

We investigated around 25,000 sets of such gradually changing F_i and F_{i+1} belonging to the frames of self captured videos. In [16] we recommended its value as 0.52, but after further experiments, we recommend its value as 0.47.

$DISP_{diff}$: Considering abruptly changing successive frames, say F_i and F_{i+1} (F_i and F_{i+1} are entirely/significantly different frames and we can call abrupt change of scene in these frames), this threshold represents ratio between count of 8×8 blocks of F_i not present in adjacent frame F_{i+1} and count of 8×8 blocks in a video frame.

We investigated approximately 23,000 sets of such abruptly/significantly changing F_i and F_{i+1} belonging to the frames of self captured videos. In [16] we recommended its value as 0.79, but after further extensive experiments, we recommend its ($DISP_{diff}$) value as 0.74.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm: $\boldsymbol{blkDisp(F_C, F_N)}$ {Computes and return count of 8×8 blocks of F_C not present in F_N}

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

```
begin
  initialize k ← 0, mC ← 0, row ← h/8 , col ← w/8
  for i ← 1 to row do
     for j ← 1 to col do
        entC[k][0] ← ENT(F_C(i,j))
        entN[k][0] ← ENT(F_N(i,j))
        {F_C(i,j) is one of the 8 × 8 blocks of F_C indexed at i, j and ENT returns the Entropy of a
        block}
        entC[k][1] ← i
        entC[k][2] ← j
        entN[k][1] ← i
        entN[k][2] ← j
        k ← k + 1
     end for
  end for
  for i ← 1 to bCount do
     mark[ i ] ← 0
  for i ← 1 to bCount do
     for j ← 1 to bCount do
        v_1 ← m_1 × entN[j][0]
        v_2 ← m_2 × entN[j][0]        {m_1 and m_2 are the margins and need to be tuned}
        if v_1 ≥ entC[i][0] and v_2 ≤ entC[i][0] then
           check ← MSE(F_C(entC[i][1], entC[i][2]), F_N(entN[j][1], entN[j][2]))
           {MSE returns Mean Squared Error between 8 × 8 block of F_C and 8 × 8 block of F_N}
           if check ≤ bThr and mark[i] = 0 then    {bThr is a threshold}
              mC ← mC + 1 , mark[i] ← 1
           end if
        end if
     end for
  end for
  return (bCount − mC)
end    {End of the algorithm/scheme blkDisp}
```

Figure 2: Scheme blkDisp to compute the count of 8×8 *blocks in a frame* (F_C) *not present in another frame* (F_N)

3. ABRUPT SCENE CHANGE DETECTION

In this section, we present the dual threshold based schemes (1) $SceneChange\_S1$ and (2) $SceneChange\_S2$ to detect the abrupt scene change. Besides the static thresholds presented in previous section, in these schemes we have used an additional feature MSE to form two dynamic thresholds, MSE_{high} and MSE_{global}. We compared (using MSE) all the successive frames F_i and F_{i+1} in a given video and computed $m-1$ MSEs, where m is the count of frames in the given video). These $m-1$ MSEs have been used to compute the dynamic thresholds, MSE_{high} and MSE_{global}. Values of these thresholds are computed dynamically for individual videos and are defined as follows:

(a) MSE_{global}: It is one of the dynamic thresholds (we call this threshold as the global MSE) and computed (Equation 2) as the average of all MSEs between each successive frames (F_i and F_{i+1}) in a given video.

$$MSE_{global} = \frac{\sum_{i=1}^{m-1} D(i)}{m-1} \qquad (2)$$

where, m is the count of frames in the given video and $D(i)$ is the vector which stores the values of $m-1$ MSEs (here, each MSE is computed by comparing i^{th} and $(i+1)^{th}$ frames in the given video).

(b) MSE_{high}: It is another dynamic threshold and computed (Equation 3) as the average of all MSEs between each successive frames (F_i and F_{i+1}) which are greater than or equal to the global MSE (MSE_{global}).

$$MSE_{high} = \left(\frac{\sum_{i=1}^{m-1} D(i)}{t}\right) \qquad (3)$$

$$if\ D(i) \geq MSE_{global}$$

where, m is the count of frames in the given video; $D(i)$ is a vector which stores the values of $m-1$ MSEs (here, each MSE is computed by comparing i^{th} and $(i+1)^{th}$ 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}.

These static and dynamic thresholds have been used in the schemes $SceneChange\_S1$ and $SceneChange\_S2$ to find the abrupt change of scenes in a video. Both schemes are almost same but some of the steps of the scheme $SceneChange\_S1$ are not performed in the scheme, $SceneChange\_S2$. Both schemes have been jointly presented in Figure 3 and discussed subsequently.

Following are the broader steps of both the schemes presented in Figure 3: (1) computation of the MSEs by comparing successive frames (F_i and F_{i+1}), (2) computation of the dynamic thresholds, MSE_{high} and MSE_{global} using Equation 2 and Equation 3, and (3) selection and rejection of the successive frames to be considered as the possibility of the abrupt scene change. The third step is done in iterations where selection and rejection have been decided with static and dynamic thresholds. Here, we re-examined those successive frames whose differences (computed using MSE) are lying at the boundaries (lower and higher, tuned using margin parameter or variable) of the dynamic threshold, MSE_{high}.

In these schemes, we rejected (definite rejection) all such successive frames (F_i and F_{i+1}) and did not consider as the possibility of the abrupt scene change (between F_i and F_{i+1}) whose differences (computed using MSE) are lesser than the dynamic threshold, MSE_{global}. We also rejected (did not consider as the possibility of the abrupt scene change) all such successive frames (F_i and F_{i+1}) whose differences (computed using MSE) are lesser than $m_1 \times MSE_{high}$ (where m_1 is a margin parameter ($m_1 < 1$) and needed to be tuned).

We considered those successive frames (F_i and F_{i+1}) as probable candidates to be re-examined for the abrupt scene change (between F_i and F_{i+1}) whose differences are in the range between $m_1 \times MSE_{high}$ and MSE_{high}, where, $m_1 < 1$ is a margin parameter and to be tuned by the user. We also considered those successive frames (F_i and F_{i+1}) as probable candidates to be re-examined for the abrupt scene change (between F_i and F_{i+1}) whose differences are in the range between $MSE_{high}$and $m_2 \times MSE_{high}$, where, $m_2 > 1$ is a margin parameter and to be tuned by the user. In other words, we re-examined those successive frames which are lying at the boundaries (lower and upper boundaries using margin parameters m_1 and m_2 respectively) of the threshold, MSE_{high}.

While re-examining the successive frames whose differences (computed using MSE) are lying at the lower and upper boundaries of the threshold MSE_{high}, we included those successive frames (F_i and F_{i+1}) as the probable candidate for abrupt scene change, where the ratio between count of displaced 8×8 blocks between F_i and F_{i+1} and count of total blocks in the video frame F_i is greater than or equal to the static threshold $DISP_{diff}$. If, we found any such successive frame, then we increased the value of a variable $temp$ by 1, where $temp$ was initially initialized to 0. Considering the successive frames whose differences (computed using MSE) are more than $m_2 \times MSE_{high}$ have been further re-examined in the scheme, $SceneChange\_S1$. Here, we rejected such successive frames (F_i and F_{i+1}) as the probable candidate for abrupt scene change, where the ratio between count of displaced 8×8 blocks between F_i and F_{i+1} and count of total blocks in the video frame F_i is less than or equal to $DISP_{gradual}$. To reduce the computational time, we did not perform this step in the scheme, $SceneChange\_S2$.

As shown in Figure 3, if $temp \geq 1$, then in both the schemes ($SceneChange\_S1$ and $SceneChange\_S2$) we reported the abrupt scene change in all the successive frames which were earlier identified as the probable candidates for abrupt scene change. Otherwise, in the scheme, $SceneChange\_S1$, we repeated the process by making MSE_{global} as MSE_{high} and recomputed the MSE_{high} for the newly computed MSE_{global}. We repeated these steps until $temp \geq 1$ or there is no such successive frames left in the video, where the MSE between those frames are greater than MSE_{global}.

Hence, in the scheme $SceneChange\_S1$, we kept on dynamically changing the thresholds MSE_{global} and MSE_{high} depending on the actual count of abrupt changes in a given video. If, 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 all successive frames having MSE in the described range have been explored.

Algorithms: $\boldsymbol{SceneChange\_S1(V)}$ **and** $\boldsymbol{SceneChange\_S2(V)}$

m	{Count of frames in video, **V**}
$count$	{A variable: stores the count of scenes in video, **V**}
$D[1..m-1]$	{A vector: stores the MSE between successive frames in **V**}
$mark[1..m-1]$	{A vector: stores Boolean value}
$blkCount$	{It is the count of 8×8 blocks in a frame of video, **V**}

begin

initialize $count \leftarrow 0, i \leftarrow 1, temp \leftarrow 0, t \leftarrow 0, var \leftarrow 0$
for $i \leftarrow 1$ to $m-1$ do
  $D[i] = MSE(V(i), V(i+1))$ { MSE returns the MSE between i^{th} and $i+1^{th}$ frames}
  $var \leftarrow var + D[i]$
end for
$MSE_{global} = var/(m-1)$
{MSE_{global} is computed as the average MSE of all the successive frames in a video}

LABEL 1 **{A label to repeat the steps}**

$var \leftarrow 0$
for $i \leftarrow 1$ to $m-1$ do
  if $D[i] \geq MSE_{global}$ then { MSE of successive frames is checked with global MSE }
    $var \leftarrow var + D[i]$,
    $t \leftarrow t+1$
  end if
end for
$MSE_{high} = var/t$
{MSE_{high} is computed as average MSE of such successive frames for which $MSE \geq MSE_{global}$ }
for $i \leftarrow 1$ to $m-1$ do
    $mark[i] \leftarrow 0$
end for
$temp \leftarrow 0$

contd..

```
for i ← 1 to m − 1  do
        p_1 ← m_1 × MSE_high,
        p_2 ← m_2 × MSE_high
        {m_1 < 1 and m_2 > 1 are the margins and need to be tuned}
        if D[i] ≥ p_1 and D[i] < MSE_high  then
            if blkDisp(V(i), V(i + 1))/ blkCount ≥ DISP_diff  then
                mark[i] ← 1,
                count ← count + 1,
                temp ← temp + 1
            end if
        end if
        if D[i] ≤ p_2 and D[i] > MSE_high  then
                if blkDisp(V(i), V(i + 1))/blkCount ≤ DISP_diff  then
                        continue the loop
    else
                    mark[i] ← 1,
                    count ← count + 1,
                    temp ← temp + 1
                end if
        end if
    if D[i] > p_2  then
                if blkDisp(V(i), V(i + 1))/blkCount ≤ DISP_gradual  then continue the loop
                {This comparison is not performed in the scheme, SceneChange_S2}
                mark[i] ← 1,
                count ← count + 1
    end if
end for
if temp < 1
        MSE_global ← MSE_high,
        t ← 0,
        count ← 0
        go to LABEL 1                {Scheme, SceneChange_S2 does not repeat this step}
end if
t ← 1
for i ← 1 to m − 1  do
        if mark[i] = 1  then
                SChange[t] ← i     {SChange is a vector of length count + 1 and stores the frame
                t ← t + 1          indices of such frames in V after which abrupt change is detected}
        end if
end for
SChange[t] ← count + 1          {Last field of SChange stores the count of scenes in video, V}
return SChange
end                             {End of scheme SceneChange_S1}
```

Figure 3: Proposed schemes $SceneChange\_S1$ and $SceneChange\_S2$ to detect the abrupt change of scene

Journal of Theoretical and Applied Information Technology
31st July 2018. Vol.96. No 14

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

However, to reduce the computational time, we did not perform the step to repeat the process of re-computation of the dynamic thresholds MSE_{global} and MSE_{high} in the scheme, $SceneChange\_S2$ and reported the abrupt scene change in all the successive frames which were earlier identified as the probable candidates for abrupt scene change. In other words, MSE_{global} and MSE_{high} have been computed once and accordingly we decided whether there is abrupt scene change in the successive frames or not. Certainly, the applied less efforts (compared with the scheme, $SceneChange\_S1$) due to non inclusion of some of the steps will reduce the required computation time, but may wrongly detect or wrongly reject some of the successive frames having abrupt scene change.

4. EXPERIMENTAL ANALYSIS

Based on conducted experiments, this section presents the analysis of the performance of the schemes presented in previous section along with the experimental details.

Experiments have been conducted using 50 self captured videos. Frames in each video are of the size 352×288 pixels (*i.e.* cif video). We manually counted the number of scenes (*i.e.* the abrupt change of scene between two successive frames) in these videos. Table 1 consolidates the number of scenes in these videos. These videos are of duration 1 to 10 minutes and having abrupt change of scene ranging between 9 and 74.

Table 1: Briefing of abrupt change of scene in each video

Video #	Corresponding count of abrupt change of scene
1 to 10	28, 62, 49, 32, 42, 31, 16, 46, 27, 31
11 to 20	20, 41, 23, 9, 58, 47, 21, 23, 26, 35
21 to 30	31, 49, 36, 52, 59, 71, 47, 43, 56, 39
31 to 40	67, 53, 61, 41, 34, 33, 38, 59, 69, 73
41 to 50	55, 53, 41, 23, 66, 56, 74, 72, 63, 26

Out of 50 videos, some videos have gradually changing scenes before an abrupt change of scene between consecutive frames encountered; some of them are with fast moving objects whereas, some of them are with least object motion before the abrupt scene change encountered.

We conducted experiments to identify all the abrupt change of scene (using both the schemes presented in previous section) in each of the 50 videos detailed earlier.

Performance of the proposed schemes, $SceneChange\_S1$ and $SceneChange\_S2$ have been evaluated according to correctly identified abrupt change of scenes, along with incorrectly identified abrupt change of scene and incorrectly rejected abrupt change of scene. Based on the experimental observations, Figure 4 presents the plot between

Actual – Actual represents the actual count of abrupt change of scenes in each video. We manually counted it for each of the 50 videos.

True Positive (TP) – It represents the **c**ount of correctly detected abrupt change of scenes in a video.

False Positive (FP) – It represents the count of incorrectly detected abrupt change of scenes in a video. Manually we observed that there is no abrupt change of scene between any two consecutive frames under consideration in a video, but the proposed scheme incorrectly identified the abrupt change between those consecutive frames under consideration, *i.e.* single scene between i^{th} frame and $(i + j)^{th}$ frame of the video is wrongly identified as two or more scenes.

False Negative (FN) – It represents the count of incorrectly rejected abrupt change of scene in a video.

As seen from Figure 4, proposed scheme $SceneChange\_S1$ detected all the abrupt change of scenes (9 such abrupt change of scenes are there in this video) in the video #9, whereas, it could not detect one abrupt change of scene in video #1, #4, and #7 out of 28, 32, and 16 abrupt change of scenes respectively. The scheme performed worst with video #34, where 33 abrupt change of scene out of 41 were detected as abrupt scene change.

Apart from successful identification of scene change, the scheme shows tolerable percentage of false positive, FP (*i.e.* incorrectly detected abrupt change of scene), best with video #1, #7, #9, *etc.* where, not a single scene has been incorrectly detected as abrupt scene change, whereas, the worst is with video #33, where 6 out of 61 scenes has been incorrectly detected as abrupt scene change.

In all, the rate of true positive (a ratio between true positive (TP) and the total count (actual count) of abrupt change of scene in a video) for all the 50 videos is between 80% and 100% with the average (of all the 50 videos) rate of true positive as around 89%.

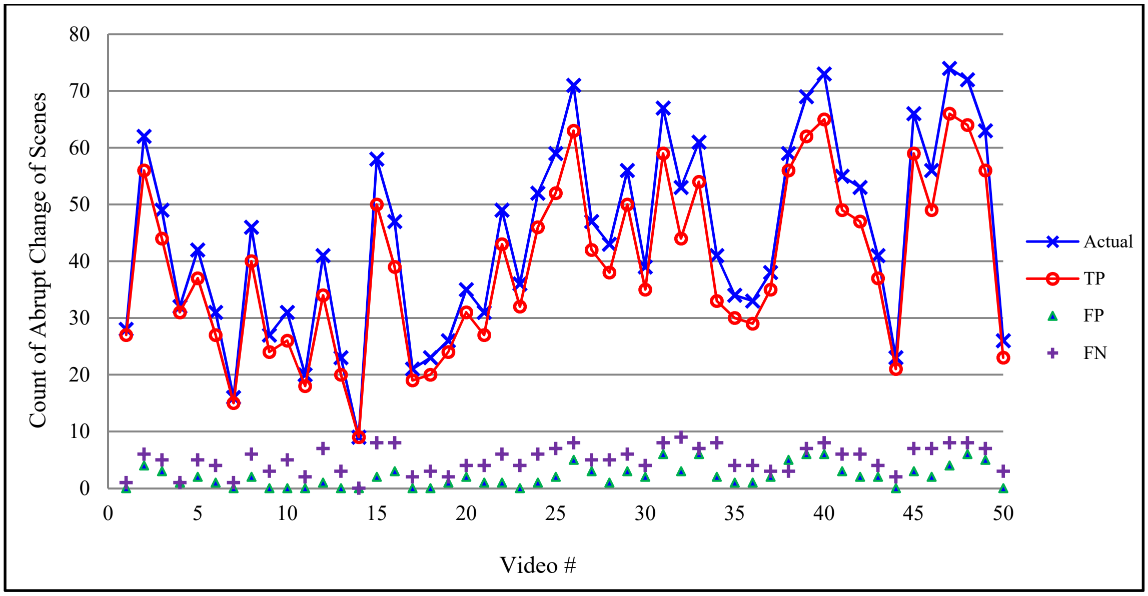

Figure 4: Performance of the proposed scheme $SceneChange\_S1$ to detect the abrupt change of scenes in a video

Similarly, Figure 5 plots the performance (Actual, TP, FP, and FN) observed with the proposed scheme $SceneChange\_S2$ to detect the scenes in the given videos (50 videos).

As observed from Figure 5, proposed scheme $SceneChange\_S2$ showed the rate of true positive for all the 50 videos in between 74% and 93% with the average (of all the 50 videos) rate of true positive as around 84%.

Besides the rate of true positive, performance of both the proposed schemes have also been evaluated using recall and precision parameters as follows:

Recall – It is computed as the ratio between count of correctly detected abrupt change of scenes in a video and count of total abrupt change of scenes in that video.

Precision: It is computed as the ratio between count of correctly detected abrupt change of scenes in a video and count of total abrupt change of scenes detected by the proposed scheme in that video.

The recall rate achieved by the proposed scheme $SceneChange\_S1$ for all the 50 videos is between 80% and 100% with the average (of all the 50 videos) recall rate as around 89%.

Further, the precision rate achieved by the proposed scheme $SceneChange\_S1$ for all the 50 videos is between 90% and 100% with the average (of all the 50 videos) precision rate as around 96%.

The recall rate achieved by the proposed scheme $SceneChange\_S2$ for all the 50 videos is between 74% and 93% with the average (of all the 50 videos) recall rate as around 84%. Further, the precision rate achieved by the proposed scheme $SceneChange\_S2$ for all the 50 videos is between 84% and 100% with the average (of all the 50 videos) precision rate as around 92%.

Based on the discussed experimental analysis, performance of the proposed scheme $SceneChange\_S1$ in terms of achieved accuracy (TP, FP, F,N, recall and precision) is better than performance of the proposed scheme $SceneChange\_S2$. However, better accuracy is achieved at cost of the required computational time which is significantly more than the time required by the proposed scheme, $SceneChange\_S2$.

Comparative Analysis: As discussed in Section 1, splitting the videos into different scenes is the necessary requirement in many applications, resulting into development of several schemes by the research community to detect the abrupt scene change. These schemes [5][7][9-10][14][15] had been developed to fulfil different requirements, *viz.* efficient detection (*i.e.* schemes which detect the scene change in lesser time), accurate detection (*i.e.* schemes which accurately detect the scene change). Some of the schemes discussed in Section 1 had been developed around the feature histogram and efficiently identify the abrupt scene change in videos; however, accurate detection of abrupt scene change using these schemes is debateable.

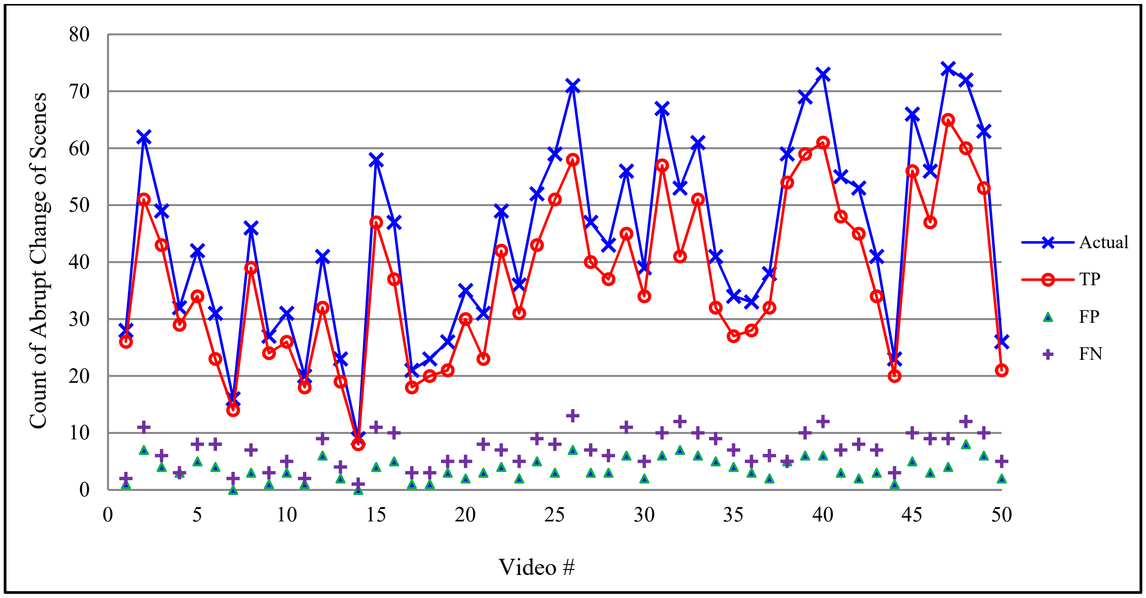

Figure 4: Performance of the proposed scheme $SceneChange\_S2$ to detect the abrupt change of scenes in a video

Comparing to these schemes, the schemes proposed in this paper fulfil both requirements simultaneously. Requirement of lesser computational time is fulfilled in the scheme, $SceneChange\_S2$, whereas, requirement of accurate detection is fulfilled in the scheme, $SceneChange\_S1$.

5. CONCLUSION

In this paper, we proposed schemes to detect the abrupt scene change in a given video. The proposed schemes used static and dynamic thresholds defined over the features MSE and DISP. Performance of the proposed schemes have been analyzed over 50 self captured videos. The true positive to identify the abrupt scene change is observed between 80% and 100% using the scheme, $SceneChange\_S1$, whereas, it is observed between 74% and 93% using the scheme $SceneChange\_S2$.

Achieved accuracy by the scheme, $SceneChange\_S1$ is better than the scheme, $SceneChange\_S2$, but it is at the cost of required processing time. Similar observations have been made in terms of the precision rate (average precision rate is observed around 96% for the scheme, $SceneChange\_S1$, whereas it is around 92% for the scheme $SceneChange\_S2$) and recall rate (average recall rate is observed around 89% for the scheme, $SceneChange\_S1$, whereas it is around 84% for the scheme $SceneChange\_S2$).

REFRENCES

[1] Michael Rubinstein, “Analysis and Visualization of Temporal Variations in Video,” Ph. D. Thesis, Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Feb 2014

[2] M. K. Thakur, “Tampered Videos – Detection and Quality Assessment,” Ph. D. Thesis, Jaypee Institute of Information Technology, June 2014

[3] Z. N. Li and M. S. Drew, “Fundamentals of Multimedia,” Pearson Education

[4] A. Bovik, “The Essential Guide to Video Processing,” Second Edition, Academic Press, Elsevier, 2009

[5] A. Vashistha, R. Nallusamy, and S. Paul, “2PASCD: An efficient 2-Pass Abrupt Scene Change Detection algorithm,” in *proc. International Conference on Multimedia Information Networking and Security*, 2010, pp. 44-48, DOI 10.1109/MINES.2010.17

[6] W. Wang and H. Farid, “Exposing Digital Forgeries in Video by Detecting Duplication,” in *proc. 9th Workshop on Multimedia & Security (MM&Sec 07*), Sept 20-21, 2007, pp. 35-42

[7] J. Yu and M. D. Srinath, “An efficient method for scene cut detection,” Pattern Recognition Letters, 22 (2001), pp. 1379-1391

[8] Wing-San Chau, O. C. Au, Tak-Song Chong, Tai-Wai Chan, and Chi-Shun 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, DOI: 10.1109/ICICS.2005.1689135

[9] H. Jiang, A. Helal, A. K. Elmagarmid, and A. Joshi, "Scene change detection techniques for video database systems," Multimedia Systems, Vol. 6, 1998, pp. 186–195

[10] Y. Yusoff, W. Christmas, and J. Kittler, "Video Shot Cut Detection Using Adaptive Thresholding," in *Proc. British Mission Vision Conference* (BMVC 2000), pp. 11-14

[11] 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. DOI: 10.1155/2010/864123

[12] M. Kobayashi, T. Okabe, and Y. Sato, "Detecting Forgery From Static-Scene Video Based on Inconsistency in Noise Level Functions," IEEE Trans. on Information Forensics and Security, Vol. 5, No. 4, December 2010, pp. 883-892 DOI: 10.1109/TIFS.2010.2074194

[13] J. Meng, Y. Juan, and S. F. Chang, "Scene Change Detection in a MPEG Compressed Video Sequence," in *Proc. IS&T/SPIE Symposium*, Vol. 2419, Feb. 1995, pp. 1-12

[14] http://compression.ru/video/quality_measure/metric_plugins/dfm_en.htm, Accessed 10 Mar 2018

[15] N. I. Radwan, N. M. Salem, and M. I. El Adawy, "Histogram Correlation for Video Scene Change Detection," in *proc. Advances in Computer Science, Engineering & Applications, Advances in Intelligent and Soft Computing*, Vol. 166, 2012, pp. 765-773

[16] M. K. Thakur, V. Saxena, and J. P. Gupta, "Learning based no reference algorithm for dropped frame identification in uncompressed video," in *Proc. Information Systems Design and Intelligent Applications, Advances in Intelligent Systems and Computing*, vol. 435, Jan 8-9, 2016, pp.451-459