Colour Histogram and Modified Multi-layer Perceptron Neural Network based Video Shot Boundary Detection

Dalton Thounaojam¹, Thongam Khelchandra², Thokchom Devi², Sudipta Roy³ and Khumanthem Singh²
¹Department of Computer Science and Engineering, National Institute of Technology Silchar, India
²Department of Computer Science and Engineering, National Institute of Technology Manipur, India
³Department of Computer Science and Engineering, Assam University Silchar, India

Abstract: The paper proposes a shot boundary detection technique using colour histogram difference and modified Multi-Layer Perceptron (MLP). In this the learning process in the MLP is modified as an evolutionary learning process using Genetic Algorithm (GA) in which the weights of the hidden layer and output layer of the MLP are updated by GA. Colour Histogram Differences (HD) between two consecutive frames are used for feature extraction. Four values: HD, HD₁, and-1 are used as an input for the modified MLP Neural Network where HD is the colour histogram difference between frame fi and fi+1, HD₁ is the colour histogram difference between frame fi and fi+1, and HD is the colour histogram difference between frame fi+1 and fi+2. The propose system is tested with the TRECVid 2001 and 2007 test data and it is also compared with latest algorithms and yields better results.

Keywords: Abrupt; fade-in; fade-out; dissolve; shot boundary detection; neural network; genetic algorithm.

Received February 11, 2016; accepted March 26, 2017

1. Introduction

Usage of the Internet has increased tremendously now-a-days with the increase in bandwidth and better connectivity throughout the world. This leads to the generation of huge amounts of data or information on the Internet in the form of text, audio, picture, video and multimedia contents. The access to this data or information has also increased tremendously and facing a huge challenge in extracting right information from the World Wide Web. Specially, accessing video and multimedia data from the Internet is also facing a problem with the unlimited generation of this type of data from the social, industrial and educational community. For better accessing of this type of data a better retrieval system is necessary. In case of a better Video Retrieval System, one has to understand the basic contents of the video as video data contain a huge or different amount of features [3, 8]. In order to understand the basic contents of the video, the video has to be divided into meaningful segments known as shot of a video and the whole process is known as temporal video segmentation or shot segmentation. A Shot is declared when the camera rolling starts and until the camera stops rolling. The process of finding the boundaries of shots in a video is known as Shot Boundary Detection (SBD).

SBD mainly consists of finding the two types of transitions:

1. Abrupt transition.

2. Gradual transition [13].

Abrupt transition (also known as cut transition) is the sudden change of the consecutive frames in a video which marks the scene change due to sudden release of the camera rolling. A Gradual transition is of four types fade-in, fade-out, dissolve and wipe transitions. All these gradual transition is a result of the editing effects in a video. Fade-in and fade-out is caused by the increase or decrease of the lightness value. In fade-in, a picture appears slowly from a monochromatic (usually black) frame. In fade-out, a picture slowly diminishes to an empty frame (usually black frame). Dissolve and wipe transitions is an effect due to overlapping of the current scene and the future scene. In dissolve, the overlapping is done in such a way that the current scene starts disappearing and the future scene starts appearing simultaneously. In wipes, the overlapping is done in such a way that the future scene grows with a certain pattern over the current scene until the future scene appears completely.

2. Related Works

In this Section, we briefly surveyed the state of art Shot Boundary Detection Techniques and algorithms. Temporal video segmentation is a part of scene segmentation, content based video indexing and retrieval system. In SBD, usually visual features like colour, edge, motion, etc., are extracted from each frame and the difference between consecutive frames are calculated [22]. Based on the visual feature, the
SBD can be categorised into compressed domain and uncompressed domain. DCT coefficients [9, 19] and motion vectors [1, 18, 19, 24, 25] are used for analyzing video in the compressed domain. In Xu et al. [26], DC images are extracted from the MPEG compressed domain for soccer video segmentation. Grass area ratio and the grass orientation are extracted from DC images. Motion vector is also extracted directly from MPEG compressed video and motion compensation is applied for detection of shot boundary [25]. In Bruyne et al. [4] the compressed characteristics of the H.264/AVC [5] is analyzed for shot boundary detection.

Histogram difference [14, 17], pixel difference [16] and edge change ratio [28] are some of the features used by the researchers in uncompress domain [22]. In Jadon et al. [10], a fuzzy classifier is proposed using frame features like colour histogram difference, pixel difference and edge count. The combination of colour histogram difference, motion compensation and texture features using fuzzy classifier for video segmentation is also proposed in Fang et al. [6]. In Kktun et al. [12], a fuzzy colour histogram using L*a*b* colour model is proposed for shot boundary detection. Colour histogram of the Just Noticeable Difference (JND) colour model is also used for shot boundary detection [11].

In Shen and Cao [21], a fuzzy clustering neural network approach for shot boundary detection is proposed which used pixel difference (Spatial Difference Metric (SDM)) and histogram difference (Histogram Difference Metric (HDM)) for finding similarities between frames.

In Ford et al. [7], four metrics namely histogram, pixel intensity, DCT coefficient and edge are used for shot boundary detection. In Lakshmi Priya and Rajmolic [20], colour, edge, motion and texture information are extracted using frame vectors of Walsh Hadamard Transform (WHT) kernel and WHT matrix for shot boundary detection.

In Yoo et al. [27], variance and average edge image features are used for detecting gradual transition. The average edge image is divided into 3x3 non overlapping blocks and for each block variance is calculated. Parabolic sequences are analyzed for detecting gradual transitions. In Lu and Shi [15], an inverted triangular pattern using colour histogram and Singular Value Decomposition (SVD) is proposed for gradual transition detection. In Baraldi et al. [2], colour histogram is also used to detect shot boundaries.

In DM Thounaojam et al. [23], a shot boundary detection technique is proposed using GLCM matrix and correlation.

In our propose system, colour histogram difference and MLP-GA is used for SBD. In this, colour histogram of red, green and blue frequencies of a frame is calculated and the difference between the consecutive frames are calculated which is given as an input to MLP-GA for classification. The learning process of MLP is made evolutionary using the GA process as discussed in section 6.

The paper is organized as follows: Section 3 provides a brief explanation of the feature extraction technique used. Section 4 explains briefly about the evolutionary learning of an MLP using GA. Section 5 briefly discusses about experimental results of the proposed system and finally the proposed system is concluded in section 6.

### 3. Feature Extraction

This section briefly discussed about the feature extraction technique used in our proposed system.

#### 3.1. Normalized Colour Histogram Difference

Colour or intensity histogram is a global feature used by many researchers [22]. Histogram of the red, green and blue frequencies of the frames is calculated and difference between the i^th and (i+1)^th frames in a video is calculated using normalized colour histogram difference [6, 10] as given in Equation (1).

\[
HD_i = \frac{1}{n} \sum_{j=1}^{n} \left[ \min(t_{ij}^r, t_{ij}^g, t_{ij}^b) + \sum_{j=1}^{n} \min(t_{ij}^r, t_{ij}^g, t_{ij}^b) \right] \]

Where, n is the total number of pixels in a frame. \(t_{ij}^r\), \(t_{ij}^g\) and \(t_{ij}^b\) are the number of red, green and blue pixels in the j^th bin of the i^th frame respectively. It is observed that \(HD_i\) value yields a value close to zero if the similarity between i^th and (i+1)^th is high and close to one if the similarity is low.

### 4. Evolutionary learning of an MLP using GA

Instead of using the concept of gradient descent in the training phase, we can use the GA to train the ANN and find the correct weights for the network. The steps are shown as follows:

#### 4.1. Initialization of the weights

An MLP is evolved by defining the genotype of GA as the weight list. Each weight can be represented as a binary number. So, each solution or individual will be a bit string which will represent the connecting weights of the ANN layers.

In Figure 1, values from \(a_1\) to \(a_5\) represents \(HD_i\), \(HD_{i+1}\) and \(HD_{i+2}\) respectively. Some training values of \(a_1, a_2, a_3, a_4\) are given in Table 2 where \(a_5\) is taken as 1. The value from \(X_1\) to \(X_n\) represents the possible combinations of the above parameters from \(a_1\) to \(a_5\).
4.3. Output of the Hidden Layer and the Output Layer

The outputs of the hidden neurons are calculated from a training set \( x \) using Equations (6) and (7).

\[
s_1 = \sum_{(i,j)} v_{ji} * x_{pi} \tag{6}
\]

\[
y_j = \text{sigmoid}(s_1) \tag{7}
\]

Where, \( y_j \) is the output of \( j^{th} \) hidden neuron.

The output of the output neurons are calculated using Equations (8) and (9).

\[
s_2 = \sum_{(j,k)} w_{kj} * y_j \tag{8}
\]

\[
o_k = \text{sigmoid}(s_2) \tag{9}
\]

Where, \( o_k \) is the output of \( k^{th} \) output neuron.

The function \( \text{sigmoid()} \) is a unipolar sigmoid function. These two operations of finding the output are performed for all input patterns.

Then we update the error with the Equation (10).

\[
E = \sum_{k=0} d_k - o_k)^2 \tag{10}
\]

Where, \( d_k \) is the desired output. This process is performed until all the training samples have been used.

4.4. Calculate the Fitness of the String or Solution

Fitness of the string or solution is calculated from the fitness definition given in Equation (11).

\[
\text{fitness} = \frac{(1 - E)}{N} \tag{11}
\]

Where, \( N \) is the number of patterns or training examples.

The above processes are repeated from Step 4.2 for all the strings or solutions of the population.

4.5. Selection

The string with highest fitness value is selected. If this highest fitness value is greater than a desired fitness value (= 1.00 in our application), then operation stops. The weights representing string with highest fitness value are used for testing or real operation phase.

4.6. Reproduction

The population is modified using operators, namely, crossover and mutation. The above processes from step 4.2 are repeated for many generations till a string or solution whose fitness value is greater than the desired fitness is obtained.
5. Experimental Results

- **Dataset Description**: TREC Vid 2001 and 2007 video test data set for shot boundary detection is used for experimentation of our proposed system. Table 1 shows information about the number of abrupt, fade and dissolve transitions for TREC Vid 2001 and 2007 test video used for experimentation. TREC Vid 2001 video test data are downloaded from Open Video Project and TREC Vid 2007 video test data is obtained from Netherlands Institute for Sound and Vision. All the test videos are MPEG compressed documentary videos.

Table 1. Description of TREC Vid 2001 and 2007 video test data.

<table>
<thead>
<tr>
<th>Video</th>
<th>Frames</th>
<th>Abrupt</th>
<th>Fade</th>
<th>Dissolve</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>D2</td>
<td>16586</td>
<td>42</td>
<td>31</td>
<td>73</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td>12304</td>
<td>39</td>
<td>64</td>
<td>103</td>
<td></td>
</tr>
<tr>
<td>D4</td>
<td>3389</td>
<td>98</td>
<td>55</td>
<td>153</td>
<td></td>
</tr>
<tr>
<td>D5</td>
<td>12608</td>
<td>45</td>
<td>23</td>
<td>71</td>
<td></td>
</tr>
<tr>
<td>D6</td>
<td>13648</td>
<td>40</td>
<td>42</td>
<td>85</td>
<td></td>
</tr>
<tr>
<td>BG_3027</td>
<td>49812</td>
<td>127</td>
<td>0</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td>BG_3097</td>
<td>44987</td>
<td>91</td>
<td>0</td>
<td>91</td>
<td></td>
</tr>
<tr>
<td>BG_3314</td>
<td>35800</td>
<td>42</td>
<td>0</td>
<td>42</td>
<td></td>
</tr>
<tr>
<td>BG_16336</td>
<td>2462</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>BG_28476</td>
<td>23296</td>
<td>176</td>
<td>1</td>
<td>178</td>
<td></td>
</tr>
<tr>
<td>BG_36136</td>
<td>29426</td>
<td>88</td>
<td>0</td>
<td>88</td>
<td></td>
</tr>
<tr>
<td>BG_37309</td>
<td>9639</td>
<td>11</td>
<td>0</td>
<td>11</td>
<td></td>
</tr>
<tr>
<td>BG_37770</td>
<td>15836</td>
<td>8</td>
<td>4</td>
<td>25</td>
<td>37</td>
</tr>
</tbody>
</table>

- **Discussion**: For discussion, the sample training set given in Table 2 is considered which includes four inputs and target data.

The experiment is carried with 20000 iterations and the target fitness value is set to 1. The weights of hidden layer and output layer generated by GA having highest fitness of 0.999965 are given below and are used throughout in our experiment for discussion. The fourth input of the proposed system is always taken as -1 and the target value of fade-in and fade-out are combined into fade as 0.1.

The weights of the hidden layer generated by GA:

$$v = [-7.501526, 2.513428, 9.062500, 0.625000]$$

And the weights of the output layer generated by GA:

$$w = [-3.1265265, 0.398083, -2.907715]$$

1. Frames 4188, 4189 and 4190 as shown in Figure 2 and their corresponding colour histogram difference $HD_{i+1} = 0.0126526$, $HD_i = 0.397404$ and $HD_{i+1} = 0.133278$ are considered for abrupt transition.

![Figure 2. Showing abrupt transition.](image)

2. Frames 288, 289 and 290 as shown in Figure 3 and their corresponding colour histogram difference $HD_{i+1} = 0.203042$, $HD_i = 0.212729$ and $HD_{i+1} = 0.200635$ are considered for fade-in transition.

![Figure 3. Showing fade-in transition.](image)

The output of hidden layer, $y$, is calculated using Equations (6) and (7) and it yields $y = [0.000000, 0.000003, -1.000000]$. Using Equations 8 and 9, output of the output neuron, $o$, is obtained as 1.000000, which is the target value of abrupt transition as shown in Table 2.
3. Frames 189, 190 and 191 as shown in Figure 4 and their corresponding colour histogram difference

\[ HD_{i-1} = 0.098481, \quad HD_{0} = 0.232019 \quad \text{and} \quad HD_{i+1} = 0.141383 \]

are considered for fade-out transition.

![Figure 4. Showing fade-out transition.](image)

The output of hidden layer, \( y \), is calculated using Equations (6) and (7) and it yields \( y = [0.984219, 0.010722, -1.000000] \). Using Equations 8 and 9, output of the output neuron, \( o \), is obtained as 0.149742, which is the target value of the fade transition as shown in Table 2.

4. Frames 7347, 7348 and 7349 as shown in Figure 5 and their corresponding colour histogram difference

\[ HD_{i-1} = 0.099002, \quad HD_{0} = 0.121386 \quad \text{and} \quad HD_{i+1} = 0.105327 \]

are considered for dissolve transition.

![Figure 5. Showing dissolve transition.](image)

The output of hidden layer, \( y \), is calculated using Equation (6) and (7) and it yields \( y = [0.999997, 0.0259028, -1.000000] \). Using Equations 8 and 9, output of the output neuron, \( o \), is obtained as 0.039260, which is the target value of dissolve transition as shown in Table 2.

- **Evaluation**: Recall, Precision and F1 score of our proposed system is calculated using Equations (12), (13) and (14) for the system evaluation.

\[
\text{Recall} = \frac{N_C}{N_C + N_M} \quad (12)
\]

\[
\text{Precision} = \frac{N_C}{N_C + N_F} \quad (13)
\]

\[
F1 \text{ score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (14)
\]

Where, \( N_C, N_M \) and \( N_F \) are the total number of correctly detected transitions, missed transitions and wrongly detected transitions respectively.

Table 3. Computation time of the proposed system.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Computation Time (in secs approx.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed method with 1K iteration</td>
<td>137</td>
</tr>
<tr>
<td>Proposed method with 10K iteration</td>
<td>241</td>
</tr>
<tr>
<td>Proposed method with 20K iteration</td>
<td>356</td>
</tr>
<tr>
<td>Proposed method with 40K iteration</td>
<td>528</td>
</tr>
<tr>
<td>Proposed method with 50K iteration</td>
<td>619</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>380</strong></td>
</tr>
</tbody>
</table>

For every iteration, computation time of the proposed system is provided in Table 3 which includes the computation time of GA process in each iteration. The experimental results and evaluation of our proposed system using TRECVid 2001 and 2007 dataset using Equations 12, 13 and 14 are shown in Table 5

The proposed system is compared with two systems - SBD using GLCM [23] and SBD using SVD and pattern matching [15] as shown in Table 4 and 6 respectively and the result shows that our proposed system performs better.

Table 4. Comparison of the SBD using GLCM with the proposed MLP-GA system

<table>
<thead>
<tr>
<th>Video</th>
<th>SBD using GLCM [23]</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall</td>
<td>Precision</td>
</tr>
<tr>
<td>D2</td>
<td>0.825</td>
<td>0.837</td>
</tr>
<tr>
<td>D3</td>
<td>0.822</td>
<td>0.820</td>
</tr>
<tr>
<td>D4</td>
<td>0.816</td>
<td>0.784</td>
</tr>
<tr>
<td>D6</td>
<td>0.975</td>
<td>0.929</td>
</tr>
<tr>
<td><strong>Average</strong></td>
<td><strong>0.880</strong></td>
<td><strong>0.850</strong></td>
</tr>
</tbody>
</table>

![Video samples showing different transitions](image)

Figure 6. Showing shot transitions of the examples discussed in Figures 3, 4 and 5 respectively.
Table 5. Performance evaluation of our proposed system using recall, precision and F1 score.

<table>
<thead>
<tr>
<th>Videos</th>
<th>Abrupt Transition</th>
<th>Gradual Transition</th>
<th>Dissolve</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Recall Precision</td>
<td>F1 score Recall Precision</td>
<td>F1 score</td>
</tr>
<tr>
<td>D2</td>
<td>0.905</td>
<td>0.905</td>
<td>0.600</td>
</tr>
<tr>
<td>D3</td>
<td>0.905</td>
<td>0.905</td>
<td>0.700</td>
</tr>
<tr>
<td>D4</td>
<td>0.905</td>
<td>0.905</td>
<td>0.800</td>
</tr>
<tr>
<td>D5</td>
<td>0.905</td>
<td>0.905</td>
<td>0.900</td>
</tr>
<tr>
<td>D6</td>
<td>0.905</td>
<td>0.905</td>
<td>1.000</td>
</tr>
<tr>
<td>Average</td>
<td>0.905</td>
<td>0.905</td>
<td>0.800</td>
</tr>
</tbody>
</table>

Table 6. Comparison of the SBD using SVM and pattern matching with the proposed system.

<table>
<thead>
<tr>
<th>Videos</th>
<th>SBD using SVM and pattern matching [15]</th>
<th>Proposed System</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Abrupt Transition</td>
<td>Gradual Transition</td>
</tr>
<tr>
<td></td>
<td>Recall</td>
<td>Precision</td>
</tr>
<tr>
<td>D2</td>
<td>0.905</td>
<td>0.905</td>
</tr>
<tr>
<td>D3</td>
<td>0.905</td>
<td>0.905</td>
</tr>
<tr>
<td>D4</td>
<td>0.905</td>
<td>0.905</td>
</tr>
<tr>
<td>Average</td>
<td>0.905</td>
<td>0.905</td>
</tr>
</tbody>
</table>

6. Conclusion
In this paper, a shot boundary detection using colour histogram and MLP-GA is proposed. The proposed system uses GA for generating the weights of MLP hidden layer and output layer by using a set of training data and target data. Colour Histogram feature of the frames are used as it is non-sensitive to motion [6, 10]. The proposed system is tested using TRECVID 2001 and 2007 test data and comparison with the latest algorithms shows better result.

Future works include incorporating the proposed system in shot classification and video retrieval process.

7. Acknowledgement
Sound and Vision video is copyrighted. The Sound and Vision video used in this work is provided solely for research purposes through the TREC Video Information Retrieval Evaluation Project Collection.

References


Dalton Thounaojam was born in Manipur, India. He did M.E. and Ph.D. from Anna University Coimbatore and Assam University Silchar, India in 2009 and 2017 respectively in Computer Science and Engineering and Information Technology. He is an assistant professor in the department of Computer Science and Engineering, NIT Silchar. His research interests include image processing, video shot boundary detection, fuzzy system and artificial neural network.

Thongam Khelchandra was born in Manipur, India. He received his Ph.D. and M.S. degree in Computer Science and Engineering from The University of Aizu, Japan in 2016 and 2007 respectively. He is an assistant professor in the department of Computer Science and Engineering, NIT Manipur. His main research interest includes Machine Learning, Pattern Recognition, Artificial Neural Network, Fuzzy Systems, Evolutionary Algorithm, Hybrid Intelligent System and their applications in Robotics, Image and Video Processing, Network Security.

Thokchom Jayshree was born in Manipur, India. She did B.E and M.Tech. from Manipur Institute of Technology, Manipur and National Institute of Technology Manipur, India in 2014 and 2016 respectively in Computer Science and Engineering. Her research interest includes image processing and video processing.

Sudipta Roy was born in Birbhum, West Bengal, India. He did MCA, M.Tech. and Ph.D. from BIT, Mesra, Jadavpur University, Kolkata, Assam University, Silchar in 2002, 2005 and 2010. He is a professor in the department of Computer Science and Engineering (formerly department of Information Technology), Assam University, Silchar. His research interests include digital watermarking, image processing, Data Security and Computer Networks.

Khumanthem Singh was born in Manipur, India. He did B.Tech.,
M.Tech., M.S. and Ph.D. from DEI, Agra, DU, Delhi, BITS, Pilani and IIT, Guwahati in 1986, 1992, 1994 and 2007 in Electrical Engineering, Control & Instrumentation, System Software and Image processing. He is an associate professor in the department of computer science & engineering, NIT Manipur. His research interests include digital watermarking, image processing, steganography and computer forensic.