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

DaltonThounaojam[1], Thongam Khelchandra[2], Thokchom Jayshree[2], Sudipta Roy[3], and Khumanthem Singh[2]
[1]Department of Computer Science and Engineering, National Institute of Technology Silchar, India
[2]Department of Computer Science and Engineering, National Institute of Technology Manipur, India
[3]Department 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_i,HD_{i-1} and-1 are used as an input for the modified MLP Neural Network where HD_i is the colour histogram difference between frame f_i and f_{i+1}, HD_{i-1} is the colour histogram difference between frame f_{i-1} and f_i and HD_{i+1} is the colour histogram difference between frame f_{i+1} and f_{i+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 the 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 metrices namely histogram, pixel intensity, DCT coefficient and edge are used for shot boundary detection. In Lakshmi Priya and Domnic [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).

$$HD_i = 1-\left(\frac{1}{3n}\right)\left[\sum_{j=1}^{n}\min(I_{rj}^{i}, I_{rj}^{i+1}) + \sum_{j=1}^{n}\min(I_{gj}^{i}, I_{gj}^{i+1}) + \sum_{j=1}^{n}\min(I_{bj}^{i}, I_{bj}^{i+1})\right] \quad (1)$$

Where, n is the total number of pixels in a frame. I_{rj}^{i}, I_{rj}^{i} and I_{rj}^{i} 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_3 represents HD_i, HD_{i-1} and HD_{i+1} respectively. Some training values of a_1, a_2, a_3, a_4 are given in Table 2 where a_4 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_4.

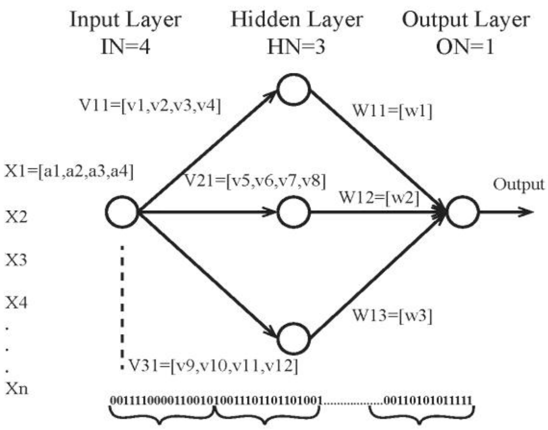

Figure 1. Structure of the MLP network.

In our application, the size of each training input is 4. The number of Hidden Neuron and Output neuron is 3 and 1 respectively. The number of total weights, *TW* is given in Equation (2).

$$TW = I * HN + HN * ON \tag{2}$$

Where, *I* is the size of input pattern, *HN* is the number of hidden neurons, *ON* is the number of output neurons.

For our application, total number of weights, *TW*=15. The genelength, *GL* is given in Equation (3).

$$GL = (B * (I * HN + HN * ON)) \tag{3}$$

Where, *B*=Number of bits per weight.
16 bits binary number is used to represent each weight, i.e., *B*=16 then genelength, *GL*= 240. So, the genotype is a 240 bits binary string.

4.2. Reconstruction of the Phenotype from the Genotype

$$y_i = \sum_{k=1}^{B} b_{ik} 2^{-k} \tag{4}$$

Where, *B* is the number of bits per weight and b_{ik} is the k^{th} bit for the i^{th} weight

$$w_i = y_i * A + B \tag{5}$$

Where, w_i is the i^{th} weight present in the string or solution, *A* is the scaling factor and *B* is the shifting factor.

In our application, value from $[-10, 10]$ is taken, so the values of *A* and *B* are taken as 20 and -10 respectively.

In this way, the weights v_{ji}, the weight from the i^{th} input to the j^{th} hidden neuron and the weights w_{kj}, the weight from the j^{th} hidden neuron to the k^{th} output neuron are calculated.

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).

$$s1 = \sum_{(i,j)} v_{ji} * x_{pi} \tag{6}$$

$$y_j = sigmoid(s1) \tag{7}$$

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

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

$$s2 = \sum_{(j,k)} w_{kj} * y_j \tag{8}$$

$$o_k = sigmoid(s2) \tag{9}$$

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

The function *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).

$$E = \frac{1}{2}\sum_{k=0}^{K}(d_k - o_k)^2 \tag{10}$$

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).

$$fitness = \frac{(1-E)}{N} \tag{11}$$

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*: TRECVid 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 TRECVid 2001 and 2007 test video used for experimentation. TRECVid 2001 video test data are downloaded from Open Video Project and TRECVid 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 TRECVid 2001 and 2007 video test data.

Videos	Frames	Transitions				Sources
		Abrupt	Fade	Dissolve	Total	
D2	16586	42	0	31	73	NASA 25th Anniversary
D3	12304	39	0	64	103	
D4	31389	98	0	55	153	
D5	12508	45	3	23	71	Airline Safety
D6	13648	40	3	42	85	Global Watcher
BG_3027	49812	127	1	0	128	TRECVid 2007
BG_3097	44987	91	0	0	91	
BG_3314	35800	42	0	0	42	
BG_16336	2462	20	0	0	20	
BG_28476	23236	176	1	1	178	
BG_36136	29426	88	0	12	100	
BG_37309	9639	11	0	8	19	
BG_37770	15836	8	4	25	37	

- *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:

$$v = \begin{bmatrix} -7.501526 & 2.513428 & 9.062500 & -0.625000 \\ -2.187805 & 0.000000 & -2.921143 & -0.468140 \end{bmatrix}$$

And the weights of the output layer generated by GA:

$$w = \begin{bmatrix} -3.1265265 & -0.390930 & -2.907715 \end{bmatrix}$$

Table 2. Training data and the Target data considered for GA.

HD_i	HD_i	HD_i	Dummy input	Target	Transition
0.2280	0.0340	0.1130	-1	1	Abrupt
0.4290	0.0380	0.1116	-1	1	
0.3420	0.3420	0.0605	-1	1	
0.4050	0.0459	0.0835	-1	1	
0.3864	0.0947	0.0562	-1	1	
0.2023	0.0986	0.1573	-1	0.1	Fade
0.1573	0.2023	0.1845	-1	0.1	
0.1845	0.1573	0.2272	-1	0.1	
0.2272	0.1845	0.1402	-1	0.1	
0.1402	0.2272	0.1829	-1	0.1	
0.1829	0.1402	0.2176	-1	0.1	
0.2176	0.1829	0.1410	-1	0.1	
0.1410	0.2176	0.1878	-1	0.1	
0.1878	0.1410	0.2288	-1	0.1	
0.2288	0.1878	0.1624	-1	0.1	
0.1058	0.1156	0.1143	-1	0.02	Dissolve
0.1143	0.1058	0.1199	-1	0.02	
0.1199	0.1143	0.0958	-1	0.02	
0.0958	0.1199	0.0989	-1	0.02	
0.0989	0.0958	0.1062	-1	0.02	
0.0831	0.0455	0.0323	-1	0.003	Normal
0.0323	0.0831	0.0324	-1	0.003	
0.0324	0.0323	0.0543	-1	0.003	
0.0543	0.0324	0.0347	-1	0.003	
0.0347	0.0543	0.0392	-1	0.003	

1. Frames 4188, 4189 and 4190 as shown in Figure 2 and their corresponding colour histogram difference HD_{i-1}=0.015522, HD_i=0.397404 and HD_i+1=0.133278 are considered for abrupt transition.

a) 4188. b) 4189. c) 4190.

Figure 2. Showing abrupt transition.

The output of the hidden layer, y, is calculated using Equations (6) and (7) and it yields y=[0.000000,0.00003,-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.

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.

a) 288. b) 289. c) 290.

Figure 3. Showing fade-in transition.

The output of hidden layer, y, is calculated using Equations (6) and (7) and it yields y=[0.999999, 0.002919, -1.000000]. Using Equations (8) and (9), output of the output neuron, o, is obtained as 0.099797, which is the target value of fade 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, HD_i=0.232019 and HD_{i+1}=0.141383 are considered for fade-out transition.

a) 189. b) 190. c) 191.

Figure 4. Showing fade-out transition.

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 $HDi\text{-}1$=0.099002, HD_i=0.121386 and HD_{i+1}=0.105327are considered for dissolve transition.

a) 7347. b) 7348. c) 7349.

Figure 5. Showing dissolve transition.

The output of hidden layer, *y*, is calculated using Equations (6) and (7) and it yields *y*=[0.999897, 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.

$$Recall = \frac{N_C}{N_C + N_M} \tag{12}$$

$$Precision = \frac{N_C}{N_C + N_F} \tag{13}$$

$$F1score = \frac{2*Recall*Precision}{Recall + Precision} \tag{14}$$

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.

Methods	Computation Time (in secs approx.)
Proposed method with 1K iteration	157
Proposed method with 10K iteration	241
Proposed method with 20K iteration	356
Proposed method with 40K iteration	528
Proposed method with 50K iteration	619
Average	**380**

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 Tables 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.

Video	SBD using GLCM [23]			Proposed System		
	Recall	Precision	F1 score	Recall	Precision	F1 score
D2	0.857	0.837	0.847	0.881	0.949	0.914
D3	0.872	0.850	0.820	0.974	0.884	0.927
D4	0.816	0.784	0.799	0.878	0.915	0.896
D6	0.975	0.929	0.951	0.975	0.929	0.951
Average	**0.880**	**0.850**	**0.864**	**0.927**	**0.902**	**0.922**

a) Fade-in.

b) Fade-out.

c) Dissolve.

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.

Videos	Transitions								
	Abrupt			Fade			Dissolve		
	Recall	Precision	F1 score	Recall	Precision	F1score	Recall	Precision	F1 score
D5	1.000	0.978	0.988	1.000	1.000	1.000	0.739	0.708	0.723
BG_3027	0.977	0.893	0.933	1.000	0.333	0.500	-	-	-
BG_3097	0.956	0.879	0.916	-	-	-	-	-	-
BG_3314	0.905	0.974	0.938	-	-	-	-	-	-
BG_16336	0.900	0.900	0.900	-	-	-	-	-	-
BG_28476	0.983	0.911	0.946	1.000	1.000	1.000	1.000	0.333	0.500
BG_36136	0.943	0.933	0.934	-	-	-	0.875	0.778	0.824
BG_37309	1.000	0.917	0.957	-	-	-	0.750	0.857	0.800
BG_37770	0.875	0.778	0.824	1.000	0.800	0.889	0.800	0.690	0.741

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

Videos	SBD using SVD and pattern matching [15]						Proposed System					
	Abrupt Transition			Gradual Transition			Abrupt Transition			Gradual Transition		
	Recall	Precision	F1 score	Recall	Precision	F1 score	Recall	Precision	F1 score	Recall	Precision	F1 score
D2	0.905	0.905	0.905	0.935	0.725	0.817	0.881	0.949	0.914	0.839	0.867	0.853
D3	0.667	0.867	0.754	0.734	0.940	0.824	0.974	0.884	0.927	0.766	0.855	0.808
D4	0.888	0.897	0.892	0.727	0.741	0.734	0.878	0.915	0.896	0.709	0.796	0.750
D6	0.950	0.974	0.962	0.844	0.927	0.884	0.975	0.929	0.951	0.822	0.902	0.860
Average	0.853	0.912	0.877	0.810	0.833	0.815	0.927	0.902	0.922	0.784	0.855	0.818

6. Conclusions

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.

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

[1] Ayele E. and Dhok S., "Motion Estimation in Video Coding using Simplified Optical Flow Technique," *The International Arab Journal of Information Technology*, vol. 13, no. 6A, pp. 770-776, 2016.

[2] Baraldi L., Grana C., and Cucchiara R., "Shot and Scene Detection Via Hierarchical Clustering for Re-Using Broadcast Video," *International Conference on Computer Analysis of Images and Patterns*, vol. 9256, pp. 801-811, 2015.

[3] Brunelli R., Mich O., and Modena C., "A Survey on The Automatic Indexing of Video Data," *Journal of Visual Communication and Image Representation*, vol. 10, no. 2, pp. 78-112, 1999.

[4] Bruyne S., Deursen D., Cock J., Neve W., Lambert P., and DeWalle R., "A Compressed-Domain Approach For Shot Boundary Detection on H.264/AVC Bit Streams," *Signal Processing: Image Communication*, vol. 23, no. 7, pp. 473-489, 2008.

[5] De Neve W., Deursen D., Schrijver D., Wolf K., and Walle R., "Using Bitstream Structure Descriptions for the Exploitation of Multilayered Temporal Scalability in H.264/AVCs Base Specification," *in Proceedings of Advances in Multimedia Information Processing*, Berlin, pp. 641-652, 2005.

[6] Fang H., Jiang J., and Feng Y., "A Fuzzy Logic Approach for Detection of Video Shot Boundaries," *Pattern Recognition*, vol. 39, no. 11, pp. 2092-2100, 2006.

[7] Ford R., Robson C., Temple D., and Gerlach M., "Metrics for Shot Boundary Detection in Digital Video Sequences," *Multimedia Systems*, vol. 8, no. 1, pp. 37-46, 2000.

[8] Hu W., Xie N., Li L., Zeng X., and Maybank S., "A Survey on Visual Content-Based Video Indexing And Retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 41, no. 6, pp. 797-819, 2011.

[9] Huang C. and Liao B., "A Robust Scene-Change Detection Method for Video Segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, no. 12, pp. 1281-1288, 2001.

[10] Jadon R., Chaudhury S., and Biswas K., "A Fuzzy Theoretic Approach for Video Segmentation Using Syntactic Features," *Pattern Recognition Letters*, vol. 22, no. 13, pp. 1359-1369, 2001.

[11] Janwe N. and Bhoyar K., "Video Shot Boundary Detection Based on JND Colour Histogram," *in Proceedings of International Conference on Image Information Processing*, Shimla, pp. 476-480, 2013.

[12] Kktun O., Gdkbay U., and Ulusoy Z., "Fuzzy Colour Histogram-Based Video Segmentation," *Computer Vision and Image Understanding*, vol. 114, no. 1, pp. 125-134, 2010.

[13] Koprinska I. and Carrato S., "Temporal Video Segmentation: A Survey," *Signal Processing: Image Communication*, vol. 16, no. 5, pp. 477-500, 2001.

[14] Lam C. and Lee M., "Video Segmentation Using Colour Difference Histogram," *Multimedia Information Analysis and Retrieval*, Berlin, pp. 159-174, 1998.

[15] Lu Z. and Shi Y., "Fast Video Shot Boundary Detection Based on SVD and Pattern Matching," *IEEE Transactions on Image Processing*, vol. 22, no. 12, pp. 5136-5145, 2013.

[16] Luo M., DeMenthon D., and Doermann D., "Shot Boundary Detection Using Pixel-To-Neighbour Image Differences in Video," *in Proceedings of TRECVID Workshop Notebook Papers*, pp. 1-6, 2004.

[17] Mas J. and Fernandez G., "Video Shot Boundary Detection Based on Colour Histogram," *in Proceedings of Notebook Papers TRECVID*, 2003.

[18] Meier T. and Ngan K., "Video Segmentation for Content-Based Coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 8, pp. 1190-1203, 1999.

[19] Meng J., Juan Y., and Chang S., "Scene Change Detection in an MPEG Compressed Video Sequence," *in Proceedings of IS&T/SPIE Symposium Proceedings*, San Jose, pp. 14-25, 1995.

[20] Priya G. and Domnic S., "Walsh-Hadamard Transform Kernel-Based Feature Vector for Shot Boundary Detection," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5187-5197, 2014.

[21] Shen S. and Cao J., "Abrupt Shot Boundary Detection Algorithm Based on Fuzzy Clustering Neural Network," *in Proceedings of 3rd International Conference on Computer Research and Development*, Shanghai, pp. 246-248, 2011.

[22] Smeaton A., Over P., and Doherty A., "Video Shot Boundary Detection: Seven Years of Trecvid Activity," *Computer Vision and Image Understanding*, vol. 114, no. 4, pp. 411-418, 2010.

[23] Thounaojam D., Roy S., and Manglem K., "Video Shot Boundary Detection using Gray Level Cooccurrence Matrix," *Indian Journal of Science and Technology*, vol. 9, no. 7, pp. 1-5, 2016.

[24] Wang D., "Unsupervised Video Segmentation Based On Watersheds and Temporal Tracking," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 8, no. 5, pp. 539-546, 1998.

[25] Wang X., Wang S., and Chen H., "A Fast Algorithm for Mpeg Video Segmentation Based on Macroblock," *in Proceedings of 4th International Conference on Fuzzy Systems and Knowledge Discovery*, Haikou, pp. 715-718, 2007.

[26] Xu P., Xie L., Chang S., Divakaran A., Vetro A., and Sun H., "Algorithms and system for Segmentation and Structure Analysis in Soccer Video," *in Proceedings of IEEE International Conference on Multimedia and Expo*, Tokyo, pp. 721-724, 2001.

[27] Yoo H., Ryoo H., and Jang D., "Gradual Shot Boundary Detection Using Localized Edge Blocks," *Multimedia Tools and Applications*, vol. 28, no. 3, pp. 283-300, 2006.

[28] Zabih R., Miller J., and Mai K., "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks," *in Proceedings of the 3rd ACM International Conference on Multimedia*, San Francisco, pp. 189-200, 1995.

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.