A Novel Hybrid Adaptive Filter to Improve Video Keyframe Clustering to Support Event Resolution in Cricket Videos

S. C. Premaratne, K. L. Jayaratne, P. Sellappan

Abstract*: Automated summary generation of sports videos poses many challenges of detecting exciting events of a game. In our research, we focus on the table of content-based video summarization for cricket videos to facilitate efficient indexing of cricket events. Initially, we have identified event boundaries accurately to detect the event start and end points. To distinguish the different types of events, we need to analyze the sequence of the camera focus area. By observing the characteristics, we have categorized the camera focus area into several categories. To overcome the challenge of low accuracy we have introduced a novel algorithm for adaptive filtering for the comparison of hue histogram. The results prove that this algorithm is sufficient for accurate image clustering and this may be used in other sports event clustering as well.

***Index Terms*: Event detection, Cricket, Video summarization, Image filtering, Adaptive filtering.**

I. INTRODUCTION

Today, society is facing many problems with handling the increasing amount of sports video from TV broadcasts [1], [2]. The essential requirement in managing video is compressing its long sequence into a more small picture through an effective summarizing process. Researchers have proposed a wide variety of techniques to take full advantage of the fact that sports videos have typical and predictable sequential structures, recurrent events, consistent features and a fixed number of camera views [3], [4]. Most of this research is focused on one type of sports video by detecting specific highlights. Unlike the traditional text-based data structures, multimedia data are complex in nature and rich in information. Therefore, new concepts like data mining and data warehousing have come into being, and people are doing much research on how to effectively handle different types of multimedia data.

The necessity for automatic summary generation methods is highlighted by the fact that the semantic value of sports footage spans short durations at irregular intervals during an event (a goal in soccer, a dismissal in cricket) [5]. A one-day cricket match can last for six hours while a soccer game can last for 90 minutes. Interesting events occur intermittently, so analysis should be done to a sequence of video frames to understand the lower level feature which relates to a particular event. A combination of video sequences can be linked to an event which is related to semantics. In cricket, for example, interesting video sequences can be the bowler run-up, batsman's stroke and the direction of travel of the ball while event semantic can be a six-hit or a four.

Multimedia data mining process consists of extracting semantic structures from a video and audios which enables summarization, browsing and indexing of the video content [6]. There has been some research on sports video annotation and summarization, especially for games such as baseball, soccer, basketball; etc. Compared to other sports, less work has been done on cricket videos probably due to the increased complexity of the game and long duration of the matches. As everyone is so busy and watching full-length cricket match is time-consuming and tedious, so it is required to summarize them effectively. So, the primary purpose of a solution to solve this problem is to adequately summarize the cricket match video along with the audio and text with the capability of searching the highlights with the users' wish (sixes, wickets, fours, catches, etc.). So, each area should be given equal priority unlike in the current methods to bring a comprehensive result. Finally, the video, audio, text components of the whole cricket match should be summarized and displayed appropriately for the user to manipulate efficiently and effectively. The essential details related to the generated highlight video are displayed to the user by using semantic data extraction.

Our research is an attempt to classify the events indicated in Fig. 1 to support automatic highlight generation. In video summarization, there are two main categories, and they are highlight based video summarization, and table of contents (TOC) based video summarization [7], [8]. A drawback in highlight-based video summarization is that it has a fixed number of highlights included thus in it static. In our approach, we intend to create a table of contents-based summarization which gives the user the flexibility to select the contents based on the user requirement. (wickets, sixes, fours, catches, run outs, etc.). Initially we five primary events which are shown in Fig. 1. The delivery and the rest of the primary events were distinguished by scoreboard detection [9]. The event boundaries to detect the delivery is done using the pitch availability and pitch zoom detection. Our approach to detect the delivery was successful, and we have managed a 74% accuracy rate for delivery detection.

Revised Manuscript Received on September 22, 2019.

S. C. Premaratne, Faculty of Information Technology, University of Moratuwa.
K. L. Jayaratne, University of Colombo Schools of Computing.
P. Sellappan, School of Science and Technology, Malaysia University of Science and Technology (MUST).

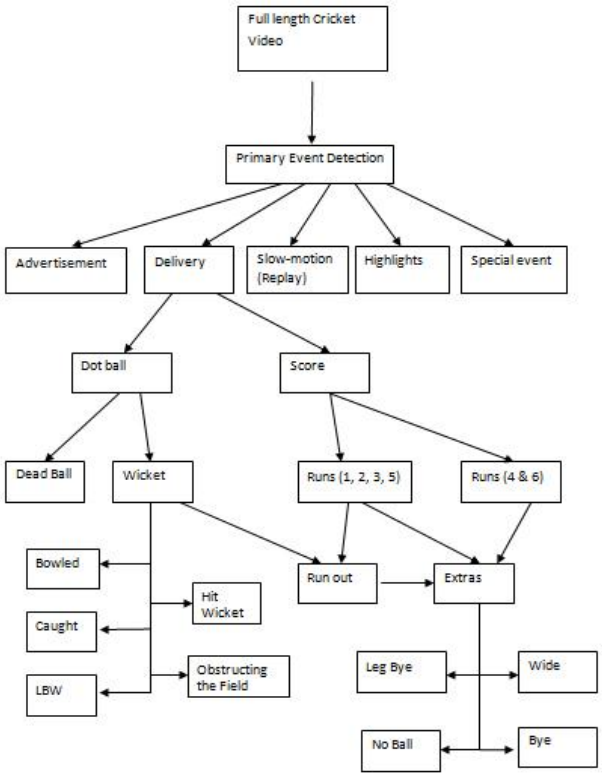


Fig. 1. Taxonomy for cricket event classification.

We have analyzed several full-length cricket videos including test matches, limited overs matches (50 overs) and T20 matches and notice that there were six different keyframe types of the video mostly highlighted throughout the duration of the entire ball. They were ground (mostly greenish), pitch (combination of the green and brown), a player from a distance (green and the color of the player), player close-up (skin color) sky (mostly a shade of blue), players gathering (multicolor) and pavilion (multicolor). We used the color histogram methodology to determine the color variation of a given keyframe, and by determining the highest percentage of color on a frame, we were able to give the attribute value for "dominant color." for the ground, pitch and sky. In the initial stage, we were unable to find the difference between a player, player close-up, and players gathering. Therefore, we investigated the possible filtering mechanisms available for accurate clustering of video keyframe types.

II. LITERATURE REVIEW

Image enhancement methods are commonly used as preprocessing steps that are applied to improve the visual quality of an image before higher level-vision tasks, such as classification and object recognition.

A. Convolutional Neural Networks (CNN)

Vivek Sharma et al. propose to extend the training of CNN based image enhancement to incorporate the high-level goal of image classification [10]. Their contribution is a method that jointly optimizes a CNN for enhancement and image classification. They have achieved this by adaptively enhancing the features on an image basis via dynamic convolutions, which enables the enhancement CNN to selectively enhance only those features that lead to improved image classification.

The proposed unified CNN architecture can emulate a range of enhancement filters with the overall goal to improve image classification in an end-to-end learning approach [11]. In addition to improving the baseline performance of vanilla CNN architectures on all datasets, their method shows promising results in comparison to the state-of-the-art using our static/dynamic enhancement filters. Also, the enhancement filters can be used with any existing networks to perform explicit enhancement of image texture and structure features, giving CNN's higher-quality features to learn from, which in turn can lead to more accurate classification. However, the data overfitting issue is always a worry for CNN, and it is reflected in this research as well.

B. Adaptive filtering for noise removal

In general, most adaptive filters are based on adaptively adjusting the parameters of the supposedly optimum filter based on the estimation of the unknown parameters. For example, adaptive Wiener or Kalman filter can be based on an on-line estimation of the signal and noise statistics from available data. These algorithms, in general, are computationally intense and expensive. With some minor assumptions, the adaptation can be made more affordable. For example, if it is assumed that the noise is an independent Gaussian random process with zero mean but unknown variance, or slowly varying variance (both in time and space), then the unknown variance can be estimated by using the residual of the filter over the last N samples. Alternatively, when the signal is not stationary, it can be assumed that it is locally stationary and the local data can be used to estimate the signal statistics; i.e., autocorrelation function.

The general problem of adaptive filters is discussed in the work by [7]. In particular, some suitable algorithms for adaptive spatial-temporal noise filtering for video images are reviewed by [2]. In this case, the adaptation is done in terms of first estimating what pixels, both in space and time, belong to an object and then only use them for averaging.

Ali M. Reza [12] has presented a first-order Kalman filter with motion detection algorithm, in a simple form. Based on the findings, recommended usage is the Kalman filter along with the motion detection algorithm. Although it is possible to use a second-order recursive filter for better performance, as it is shown in the results, the difference between the two, for a given effective averaging length, is at most in the order of 2. Adaptation and motion detection for the second order recursive filter, however, would be more complicated. The analysis is based on the assumption that the image sequence is presented to the filtering algorithm as a three-dimensional (3D) digital signal. Each frame is represented by its spatial-temporal features and sequence of frames.

The scope of this research is based on limited cases. Using what are the dominant color(s) in each frame of a video, we can determine the camera focus for each frame successfully. Combined with their attributes of the same video such as the edge percentage we can determine the camera focus with a very high accuracy rate.

III. VIDEO PROCESSING

In a cricket video, we observed that even a single delivery is captured using multiple cameras and after each camera change the angle of the video differs significantly. So, in order to analyze this change and get this feature extracted out of the videos we use we were able it implements a suitable methodology. To detect a camera change, we are analyzing every frame of a video frame sequence, and by the number of similar pixels in two or more adjacent videos, we can determine whether these two frames belong to the same camera angle of not. So, by this comparison, we can segment each video into pieces which consist of a single camera angle.

Eight frames are taken from the at a time to detect the camera changes (Fig. 2). In this case, the frame indexed of camera changes will be stored in an array. Both Short Transitions and Long Transitions are considered. As the first step, the absolute difference is taken from two frames. Then Bitwise AND operator are applied. Then a binary image is created after converting each frame into a gray image. The threshold value is 10. Then the mean of each frame is stored in an array.

Let starting frame = f_i; where i is the index of the frame in the whole video.

let mean[4] = []
let bitwiseAndFrame
let grayFrame
let binaryFrame

$af_0 = |f_{i+2} - f_{i+3}|$
$af_1 = |f_{i+2} - f_{i+4}|$

bitwiseAndFrame = af_0 & af_1
grayFrame = rgbToGray(bitwiseAndFrame)
binaryFrame = grayToBinary(grayFrame, 10)
mean[0] = mean(binaryFrame)

$af_0 = |f_{i+3} - f_{i+5}|$
$af_1 = |f_{i+4} - f_{i+5}|$

bitwiseAndFrame = af_0 & af_1
grayFrame = rgbToGray(bitwiseAndFrame)
binaryFrame = grayToBinary(grayFrame, 10)
mean[1] = mean(binaryFrame)

$af_0 = |f_{i+4} - f_{i+5}|$
$af_1 = |f_{i+4} - f_{i+6}|$

bitwiseAndFrame = af_0 & af_1
grayFrame = rgbToGray(bitwiseAndFrame)
binaryFrame = grayToBinary(grayFrame, 10)
mean[2] = mean(binaryFrame)

$af_0 = |f_{i+5} - f_{i+7}|$
$af_1 = |f_{i+6} - f_{i+7}|$

bitwiseAndFrame = af_0 & af_1
grayFrame = rgbToGray(bitwiseAndFrame)
binaryFrame = grayToBinary(grayFrame, 10)
mean[3] = mean(binaryFrame)

short_transition_threshold = 30

if((short_transition_threshold < mean[1] – mean[0] AND short_transition_threshold < mean[2] – mean[3])) {
$i+4^{th}$ frame is a camera changing point.
}

$af_0 = |f_i - f_{i+2}|$
$af_1 = |f_{i+1} - f_{i+2}|$

bitwiseAndFrame = af_0 & af_1
grayFrame = rgbToGray(bitwiseAndFrame)
binaryFrame = grayToBinary(grayFrame, 10)
mean[0] = mean(binaryFrame)

$af_0 = |f_i - f_{i+5}|$
$af_1 = |f_{i+1} - f_{i+5}|$

bitwiseAndFrame = af_0 & af_1
grayFrame = rgbToGray(bitwiseAndFrame)
binaryFrame = grayToBinary(grayFrame, 10)
mean[0] = mean(binaryFrame)

$af_0 = |f_{i+2} - f_{i+7}|$
$af_1 = |f_{i+2} - f_{i+6}|$

bitwiseAndFrame = af_0 & af_1
grayFrame = rgbToGray(bitwiseAndFrame)
binaryFrame = grayToBinary(grayFrame, 10)
mean[0] = mean(binaryFrame)

$af_0 = |f_{i+5} - f_{i+7}|$
$af_1 = |f_{i+5} - f_{i+6}|$

bitwiseAndFrame = af_0 & af_1
grayFrame = rgbToGray(bitwiseAndFrame)
binaryFrame = grayToBinary(grayFrame, 10)
mean[0] = mean(binaryFrame)

long_transition_threshold = 130

if((long_transition_threshold < mean[1] – mean[0] AND long_transition_threshold < mean[2] – mean[3])) {
i+4th frame is a camera changing point.
}

The i is incremented by one and iterate through the whole video. Then the output array contains camera changing points. Then the video is cropped from those points.

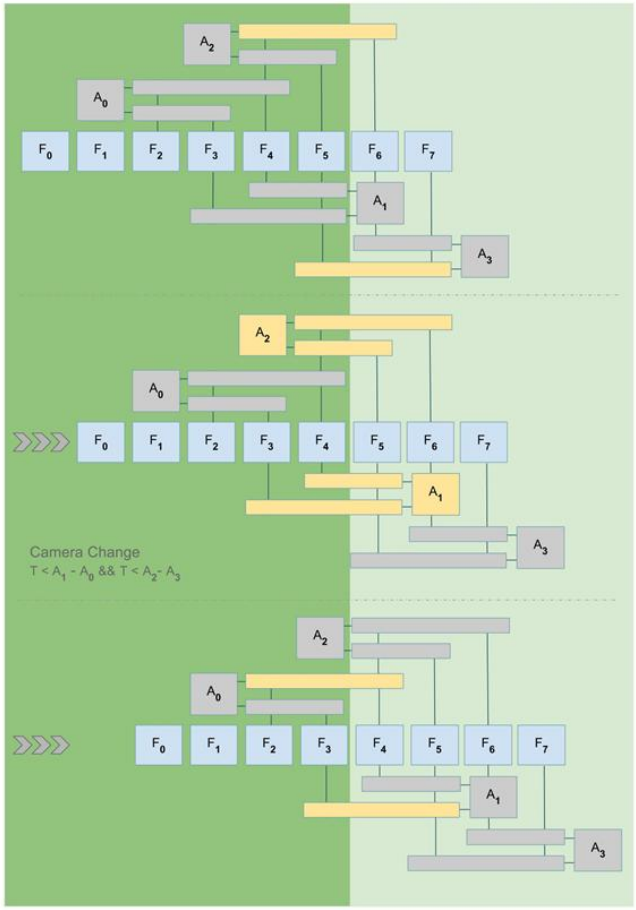


Fig. 2. Detecting Camera Changes

Classifier information

Attribute	Possible Values
Pitch Availability	1,0 (Yes, No)
Pitch Zooming	1,0 (Yes, No)
Edge Percentage	0 – 100
Camera Focus Area	0 – 5 (0 – GROUND 1 – SKY 2– PAVILION 3 – GROUND AND SKY 4 – GROUND AND PAVILION 5 – SKY AND PAVILION)
Camera Change Length (Duration)	0 – 3 (0 – LOW 1 – MID 2 - HIGH)
Camera Motion (X Direction Pattern)	0 – 12
Camera Motion (Y Direction Pattern)	0 – 12
Camera Motion (Z Direction Pattern)	0 – 12

Ball starting point Classifier

Pitch Availability
Pitch Zooming
Edge Percentage
Camera Focus Area
Camera Change Length (Duration)
Camera Motion (X-Direction Pattern)
Camera Motion (Y-Direction Pattern)
Camera Motion (Z Direction Pattern)

Ball end point classifier

Camera Focus Area
Camera Motion (X-Direction Pattern)
Camera Motion (Y-Direction Pattern)
Camera Motion (Z Direction Pattern)

Shot detection classifier (for each camera change)

Pitch Availability
Pitch Zooming
Edge Percentage
Camera Focus Area
Camera Change Length (Duration)
Camera Motion (X-Direction Pattern)
Camera Motion (Y-Direction Pattern)
Camera Motion (Z Direction Pattern)

IV. SEQUENCE ANALYSIS OF EVENTS

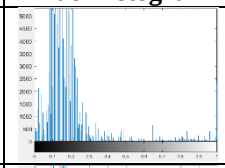
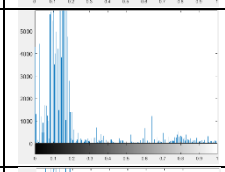
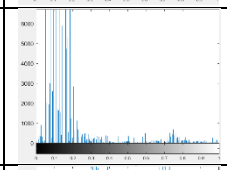
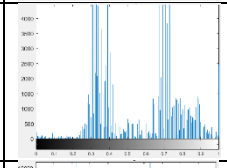
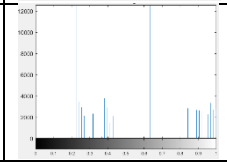
We have identified the different main types of keyframes in different types of events such as a dot ball, a boundary, a six, a dismissal, etc. the keyframe sequence for a dot ball, a boundary, a six and a dismissal are given in Fig. 3,4,5 and 6 respectively. From the different types of keyframes, we have categorized them into eight keyframe types shown in Fig. 7. In our approach, we investigate each keyframe type by obtaining the HSV color model. From the HSV color components, we have realized that the primary differentiation is possible when we extract the Hue component. The different hue components of each keyframe type are given in Fig. 7.

Figure 3. Keyframe sequence of a dot ball

Figure 4. Keyframe sequence of a four

Fig. 5. Keyframe sequence of a six

Fig. 6. Keyframe sequence of a dismissal

Detail	Picture	Hue Histogram
Pitch		
Batsman		
Fielder		
Ground & Pavilion		
Sky & Pavilion		

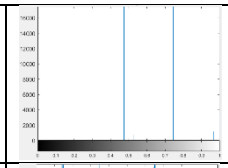
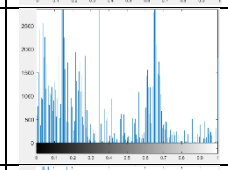
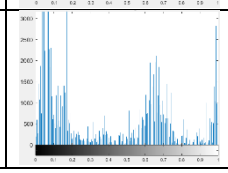
Sky		
Close-up		
Crowd		

Fig. 7. Eight different types of keyframes & Hue histogram

From the output of the hue histograms, we have observed that is it very difficult to distinguish the difference clearly. Therefore, we have applied different types of filters such as the Gaussian filter and Weiner filter to cluster the keyframe types. Since the results of applying a single filter did not achieve a considerable accuracy rate, we have attempted a hybrid approach by a combination of Gaussian filter and Weiner filter and to separately detect the keyframes.

The standard Gaussian filter is shown in equation 1.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (1)$$

Where x is the distance from the origin in the horizontal axis, y is the distance from the origin in the vertical axis, and σ is the standard deviation of the Gaussian distribution.

After doing numerous experiments on different combinations of we observed that the variance has a distinct effect on the Gaussian filter output which also helps the Weiner filters functionality. The best possible combination we found to replace the standard deviation of the Gaussian filter is by a combination of mean and variance. We take the mean value of the hue component of the keyframe and multiply it by ten then the variance of the keyframe will be divided by the mean. The code segment for this operation is in MATLAB is shown below.

```
hsv = rgb2hsv(RGB);
h = hsv(:,:,1);
m = mean2(h)*10;;
v= var(h(:))/m;
g = imnoise(h,'gaussian',v);
```

Then in the Gaussian filter the standard deviation σ is replaced by the variance v. Then the output of the Gaussian filter is taken as an input to the Weiner filter. The process of applying the standard Weiner filter is described below.

Weiner filter requires the local mean and variance around each pixel. Following two equations use neighborhoods of size M-by-N to estimate the local image mean and standard deviation. Where η is the N-by-M local neighborhood of each pixel in the image a.

μ = mean
σ = variance

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2)$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2$$

Then creates a pixel-wise Wiener filter using these estimates.

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - \nu^2}{\sigma^2}(a(n_1, n_2) - \mu) \quad (2)$$

Where ν^2 is the noise variance (equation 2), by changing the neighborhood size, we have observed that there is a significant change in the Hue histograms. After conducting several experiments, we have identified that the neighborhood size is proportionately equal to the mean intensity value of the keyframe. Therefore, we roundup the mean intensity value to the nearest integer and parsed the value to the Wiener filter function. The code segment for this operation is in MATLAB is shown below.

```
m=mean2(h);
m = round(m);
B = wienerfunc(g, [m m]);
```

After the modification to the Gaussian filter and the Weiner filter, the histogram comparison is shown in Fig. 8. From the histogram, we have gained much significant and accurate results to classify the keyframe types.
From the outcome of the histograms, we have clustered the keyframe types to the following clusters using K-means clustering.

Cluster 1: (Pitch, Batsman, Fielder)
Cluster 2: (Ground & Pavilion)
Cluster 3: (Sky & Pavilion)
Cluster 4: (Sky)
Cluster 5: (Closeup, Crowd)

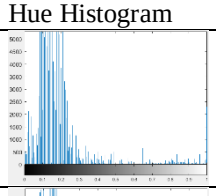
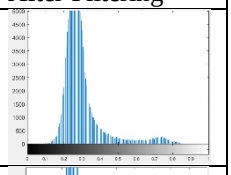
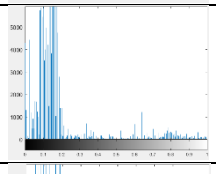
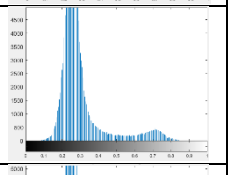
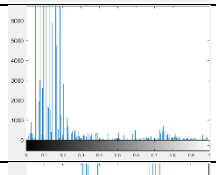
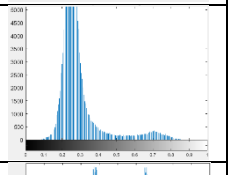
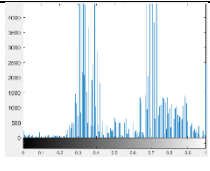
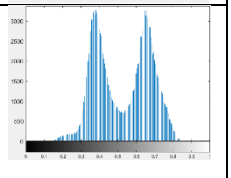
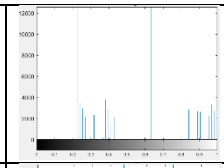
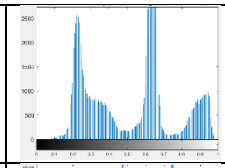
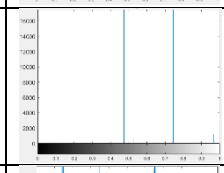
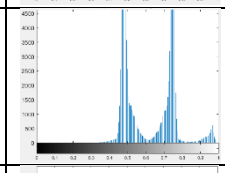
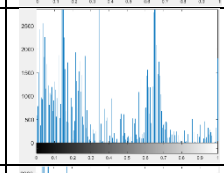
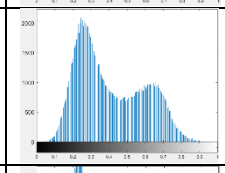
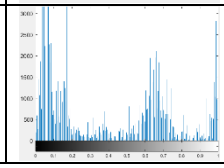
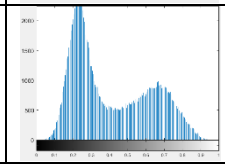
Detail	Hue Histogram	After Filtering
Pitch		
Batsman		
Fielder		
Ground & Pavilion		
Sky & Pavilion		
Sky		
Close-up		
Crowd		

Fig. 8. Histogram comparison after modified Gaussian filter

Ideally, we expected the outcome to be eight different clusters, but according to our experiments, the variation between the pitch, batsman and fielder keyframes cannot be determined as well as the difference between closeup and crowd by the hue values. Therefore, we chose the saturation component and applied the same filtering approach. From the result, we identified the three keyframe types could be distinguished by analyzing the saturation variation. The results are shown in Fig. 9.

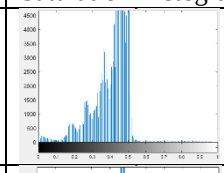
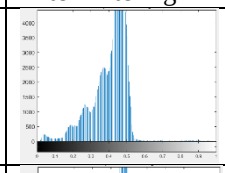
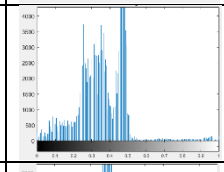
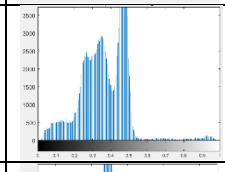
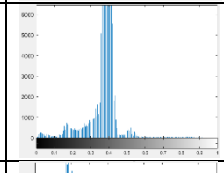
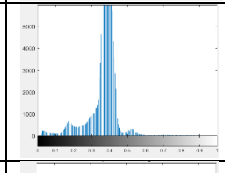
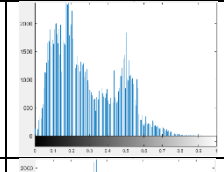
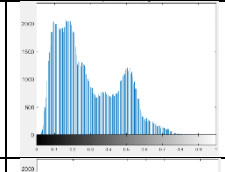
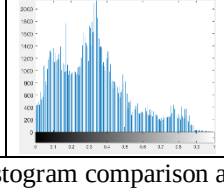
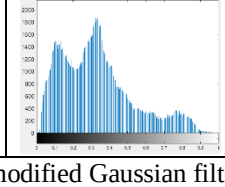
Detail	Saturation Histogram	After Filtering
Pitch		
Batsman		
Fielder		
Close-up		
Crowd		

Fig. 9. Histogram comparison after modified Gaussian filter for Pith, Batsman, Fielder, closeup, crowd keyframe types.

After the experiment results, we were able the separately identify the eight different clusters accurately.

V. EVEALUATION

Initially, we did the experiments using 50 frames for each without the filtering approach, and the result and the graph for the results are given in table I and Fig. 10 respectively.

Table I. Results of clustering without filtering.

	Pitch	Batsman	Fielder	Ground & Pavilion	Sky & Pavilion	Sky	Close-up	Crowd
Correct Recognition	21	23	18	34	36	39	26	28
Incorrect Recognition	29	27	32	16	14	11	24	22

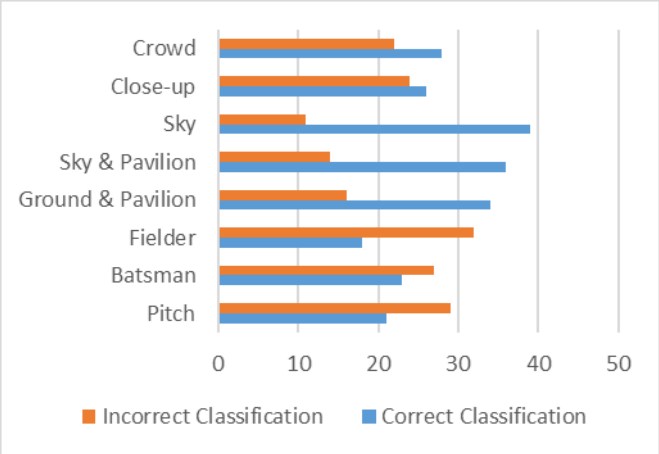


Fig. 10. The graph for Results of clustering without filtering.

From the graph, we can observe that the outcome is inadequate since there are no enhancements applied for keyframes. After applying the hybrid filtering approach, the result and the graph for the results are given in table II and Fig. 11 respectively.

Table II. Results of clustering After Hybrid Filtering.

	Pitch	Batsman	Fielder	Ground & Pavilion	Sky & Pavilion	Sky	Close-up	Crowd
Correct Recognition	34	32	27	46	48	49	32	36
Incorrect Recognition	16	18	23	4	2	1	18	14

.

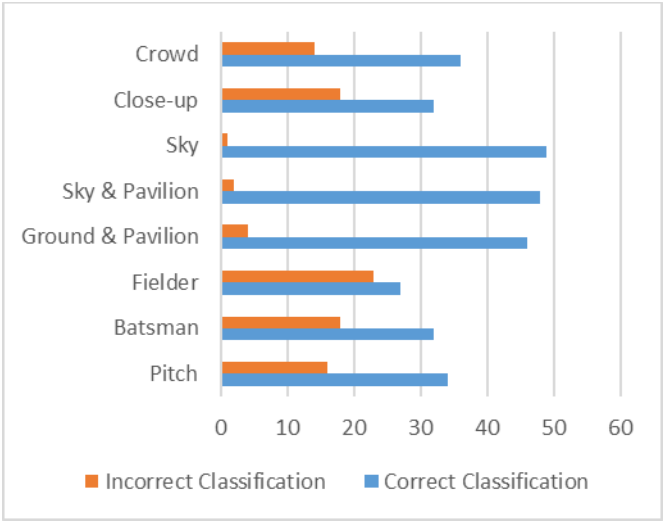


Fig. 11. Graph of Results of clustering After Hybrid Filtering.

We applied the hybrid filter to the five types of keyframes to check the accuracy using a saturation component in the HSV color model. The results and the graph is shown in table III and Fig. 12 respectively. From the results we can see clearly the accuracy of clustering has improved.

Table III. Results of clustering After Hybrid Filtering with the Saturation.

	Pitch	Batsman	Fielder	Close-up	Crowd
Correct Recognition	46	47	45	48	47
Incorrect Recognition	4	3	5	2	3

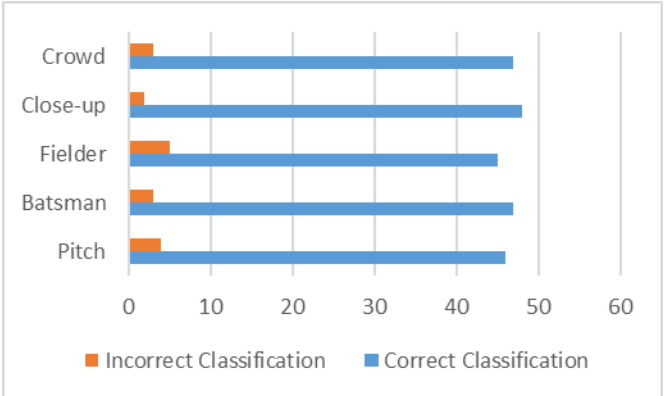


Fig. 12. Graph of Results of clustering After Hybrid Filtering with the Saturation.

After the primary dataset, we have increased the dataset by getting 150 keyframes for each type of keyframe and evaluated the filtering approach. The results are shown in table IV and Fig. 13 respectively.

Table IV. Results of clustering After Hybrid Filtering Using 150 keyframes

	Correct Recognition	Incorrect Recognition
Pitch	134	16
Batsman	136	14
Fielder	141	9
Ground & Pavilion	140	10
Sky & Pavilion	144	6
Sky	147	3
Close-up	138	12
Crowd	136	14

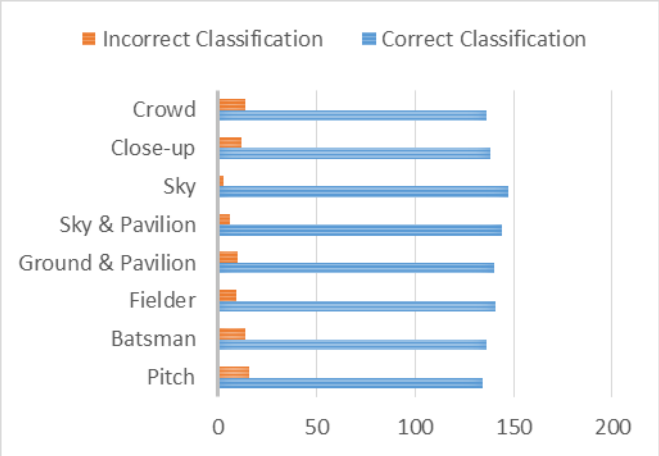


Fig. 13. Graph of Results of clustering After Hybrid Filtering Using 150 keyframes.

VI. CONCLUTION

In our work, initially, the following keyframe types were detected using the video segmentation. Pitch, Batsman, Fielder, Ground & Pavilion, Sky & Pavilion, Sky, Close-up, and Crowd. Initially, the clustering did not give enough accuracy for the clustering. Therefore, we have applied different filtering techniques the improve the clustering of keyframes. We have observed that none of the standard filtering approaches gave enough improvements for the keyframes to be recognized adequately for us to apply in the clustering step.

Therefore, we have tried a combination of the clustering approach and improved the approach be changing its variable which suited our requirement. The approach is proven to be working well for our purpose. We believe that this approach is suitable for the other sports event analysis and clustering as well.

ACKNOWLEDGMENT

This research received financial support from the University Grants Commission (UGC) of Sri Lanka under the grant number UGC/VC/DRIC/PG2017(I)/MRT/02.

REFERENCES

1. Herranz, L. and Martínez, J. M. An e-client summarization algorithm based on clustering and bitstream extraction. In Proc. of *International Conference on Multimedia and Expo*, pages 654-657. 2009.
2. Herranz, L. and Martínez, J. M. A framework for scalable summarization of video. *IEEE Transactions on Circuits and Systems for Video Technology*, 20(9):1265-1270. 2010.
3. Zhu, X., Elmagarmid, A., Xue, X., Wu, L., and Catlin, A. Insight Video: toward hierarchical video content organization for e-client browsing, summarization, and retrieval. *IEEE Transactions on Multimedia*, 7(4):648-666. 2005.
4. Zhao, Z., Jiang, S., Huang, Q., and Zhu, G. Highlight summarization in sports video based on replay detection. *In Proc. of International Conference on Multimedia and Expo*, pages 1613-1616. 2006.
5. D. W. Tjondronegoro et al., Knowledge-discounted event detection in sports video, *IEEE Transactions*, vol. 40, no. 5, pp. 1009–1024, Sep. 2010.
6. J. Shen, D. Tao, X. Li, "Modality mixture projections for semantic video event detection", *IEEE Transactions Circuits System. Video Technology*., vol. 18, no. 11, pp. 1587–1596, Nov. 2008.
7. S. E. F. De Avila, A. P. B. Lopes, A. da Luz, and A. de Albuquerque Ara´ujo, "Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.
8. M. Gygli, H. Grabner, H. Riemenschneider, and L. Van Gool, "Creating summaries from user videos," in *European conference on computer vision. Springer*, pp. 505–520. 2014.
9. S. C. Premaratne, K. L. Jayaratne, P. Sellappan. Improving Event Resolution in Cricket Videos, ICGSP'18 Proceedings of the 2nd International Conference on Graphics and Signal Processing. Sydney, NSW, Australia. Pages 69-73. 2018
10. Vivek Sharma, Ali Diba, Davy Neven, Michael S. Brown, Luc Van Gool, and Rainer Stiefelhagen. Classification Driven Dynamic Image Enhancement. CVPR 2018.
11. X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. arxiv:1609.02087, 2016.
12. ALI M. REZA, Adaptive Noise Filtering of Image Sequences in Real Time, WSEAS TRANSACTIONS on SYSTEMS, ISSN: 2224-2678 190 Issue 4, Volume 12, p 189 - 201, April 2013

AUTHORS PROFILE

Saminda Chandika Premaratne is a senior lecturer in the Department of Information Technology, Faculty of Information Technology, University of Moratuwa Sri Lanka. He obtained his B.Sc. in Computer Science from the American University of Asia in 2002 and obtained his M.Phil. in 2008 from University of Colombo School of computing. Currently, conducting his Ph.D. studies in Malaysia University of Science and Technology. His research interests are Data mining, Video processing, and multimedia information retrieval. He has published his research work in several conferences and journals. He is currently conducting research mainly on event resolution on sports videos for effective content retrieval. The goal of this research is to automate the indexing of sports videos based on the essential events that occurred in the full-length sports video. There are several postgraduate researches carried under his supervision in the areas of data mining and multimedia systems.

Dr. Lakshman Jayaratne is a Senior Lecturer attached to the University of Colombo School of Computing (UCSC), University of Colombo. Dr. Lakshman Jayaratne obtained his B.Sc (Hons) in Computer Science from the University of Colombo, Sri Lanka in 1992. He obtained his PhD in Computer Science from the University of Western Sydney, Sydney, Australia in 2006. Dr. Jayaratne is currently the principal investigator of the Automated Content Based Audio Music Monitoring Approach for Radio Broadcasting Channels in Sri Lanka, in which his team is investigating an approach to identify songs for radio channels in Sri Lanka in order to ensure the intellectual property rights of the song creators. Further, he is currently the principal investigator of the Computational Approach to Train on Music Notations for Visually Impaired in Sri Lanka where his team is using music notations to enable visually impaired musicians to overcome the barriers they had in learning music, most highlighted in reading and regenerating music notations.For his research contributions, Dr Jayaratne

was awarded Vice Chancellor's Award for "Recognition of Excellence in Research (Research Award)" in the year 2013 and 2015 at Postgraduate Convocation of University of Colombo.

Dr. Sellappan is currently Professor of IT and Dean of School of Science and Engineering and Provost of Malaysia University of Science and Technology. He holds a PhD in Interdisciplinary Information Science from University of Pittsburgh (USA), a MSc in Computer Science from University of London (UK), and a Bachelor in Statistics from University of Malaya. His research interests include Data Mining, Machine Learning, Health Informatics, Blockchain, Cybersecurity, Data Analytics and IoT.