AUTOMATIC CRICKET HIGHLIGHTS EXTRACTION USING EVENT DRIVEN AND EXCITATION BASED FEATURES

BY

Sathish Kumar S	2015103530
Sivaranjani P	2015103534
Belal R	2015103007

GUIDED BY

Dr. P. Uma Maheswari

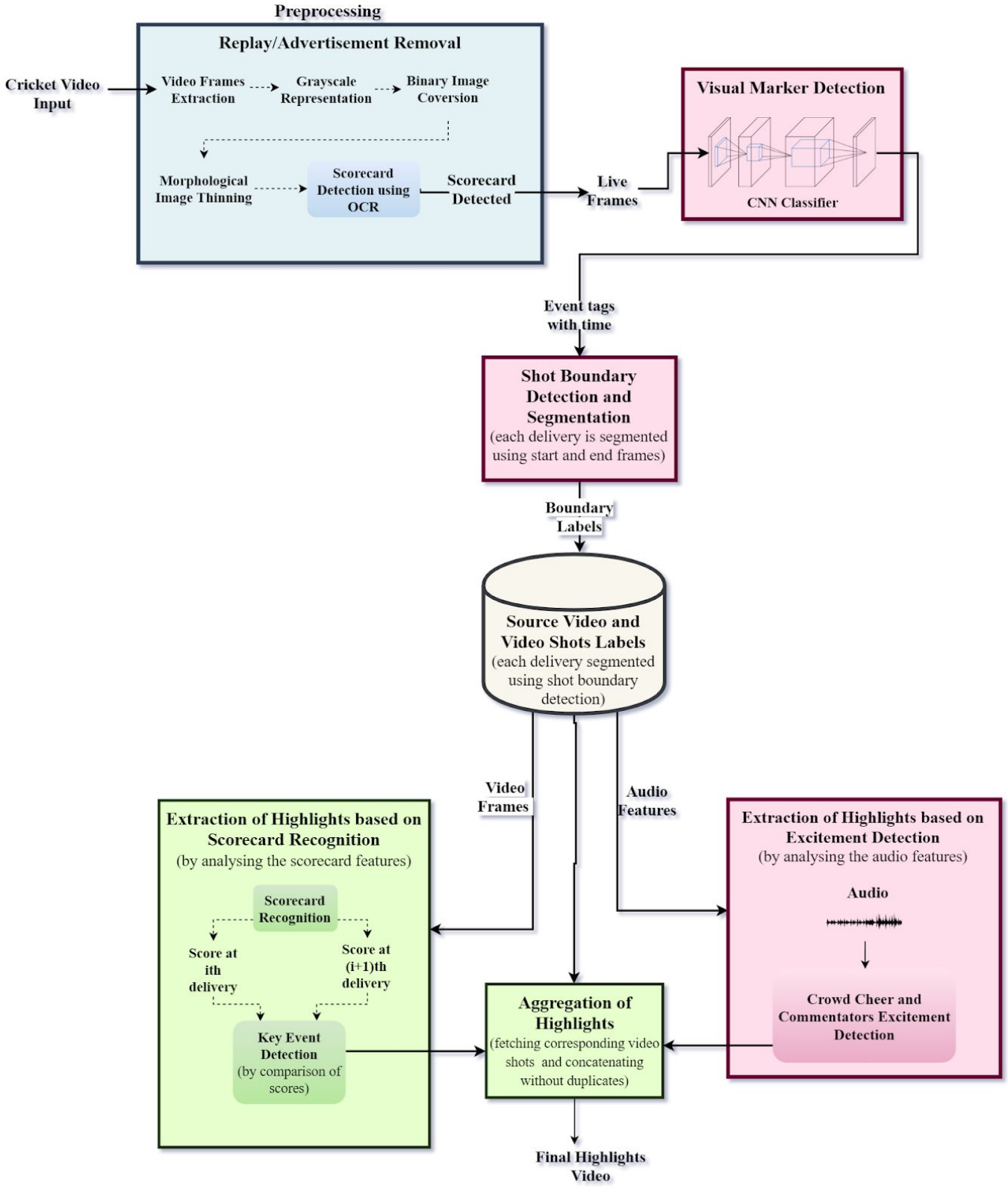
Associate Professor

Dept of Computer Science

OVERALL OBJECTIVES

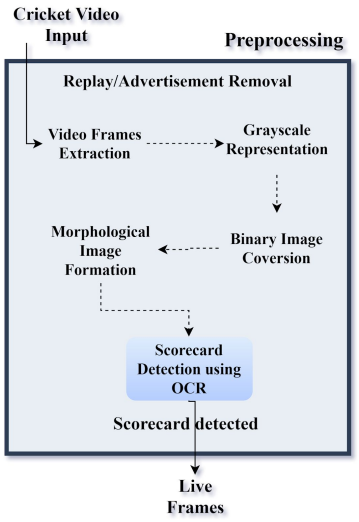
- The main objective of the project is to implement a system that automatically extracts highlights from the cricket match video.
- To remove the replays to avoid repetitive events in the highlights.
- Key events such as boundaries, wickets are identified and extracted.
- To reduce the size required to store voluminous videos.

ARCHITECTURE DIAGRAM



DETAILED MODULE DESIGN

Preprocessing- *Replay/ Advertisement removal*



This module is used for detection and removal of replays and advertisements by checking the scorecard in the video frames. Absence and presence of the scorecard are used to detect replay and live frames, respectively. The source video is converted into frames. In order to detect the scorecard by OCR, the frames are converted to grayscale image then to binary image. OCR recognizes the scorecard from the binary image. The frames with the scorecard is live frames which is given to visual marker detection along with their time.

Methodology	Optical Character Recognition (OCR)
Input	Cricket video with replays and advertisements.
Output	Video frames without replays and advertisements.

Input: Source video
Output: Live frames
Algorithm 1:

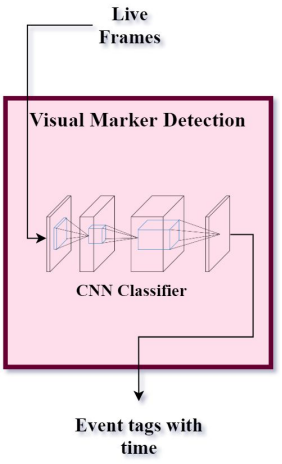
```
1)  frames[] := generate_frames(Source video)
2)  for each frame in frames[]
3)      image := grayscale(frame)
4)      image := binary_conversion(image)
5)      //comment: image is processed to detect the scorecard easily
6)      text := OpticalCharacterRecognition(image)
7)      if is_scorecard(text):
8)          mark_as_live(frame)
9)      else
10)         mark_as_replay(frame)
11)     end if
```

12) **end for**

Visual Marker Detection- *Event tagging*

This module is used for detecting events like batting, bowling, crowd, interviews, commentators etc. from the live frames using trained CNN classifier. The CNN is trained for bowling, commentators, interview, field view, crowd, stump view, umpire. The trained model is loaded to predict these classes. The CNN classifier predicts the events of the live frames. The event tag along with the time is given to the shot boundary detection and segmentation.

Methodology	CNN
Input	Live Frames
Output	Event tags with time.



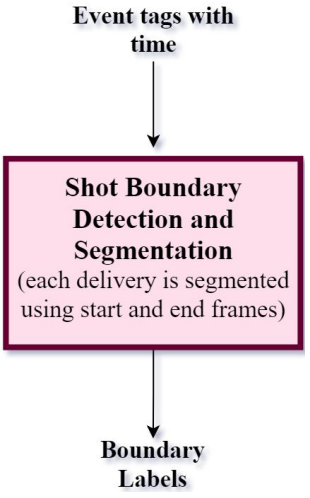
Input: Live frames
Output: Frames with event tags
Algorithm:

1) model :=load_trained_cnn_classifier()
2) **for** each frame in frames[]
3) event_tag :=model.classify(frame)
4) time := fetch_frame_time(frame)
5) out_to_CSV_file(event_tag,time)
6) **end for**

Shot Boundary Detection and Segmentation- *Video shot labels generation*

Unwanted events like crowd view, commentators, interviews, etc. are removed. Start/End frames are detected and segmented as video shots. The bowling event is considered as the start of the delivery. Every delivery is till the start of the next delivery. The frames with unwanted events are removed. Using start and end of the delivery, shots are identified. The labels containing start and end time of the delivery is stored for further use.

Methodology	Start / End frame Detection
Input	Event tags with time
Output	Labels of video shots



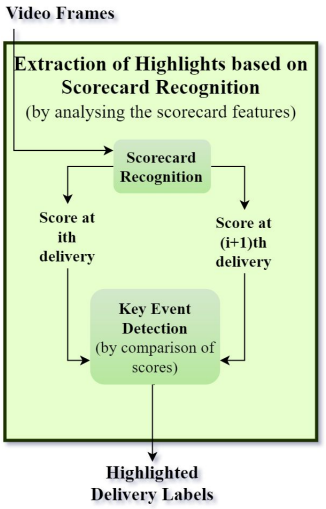
Input: event tags
Output: segmented delivery labels
Algorithm:

```
1)  i :=1 // initially ball number is 1
2)  for each event in event[]
3)      if is_bowling(event)
4)          start[i] := time(event)
5)          out.write(start)
6)          // comment: start time(boundary) is detected and appended in the output
    file
7)      else if event in [commentators, interview, crowd, umpire_signal]
8)          end[i] := time(event)
9)          out.write(end)
10)         // comment: end time(boundary) is detected and appended in the output
    file
11)     end if
12)     i+ := 1
13) end for
```

Extraction of highlights based on scorecard recognition

The difference of the score from the first frame of current shot and the first frame of the next shot is checked. If it is 4, 6 then it is added to highlights or if the wicket is increased then it is added to highlights. The score of the start frame of the delivery and the next delivery is recognized. Using the scores, key events are checked. If the score differ by more than 4, it is taken as highlights. If wicket increases, it is also considered as the highlights. Corresponding delivery labels are given as the output to the next module.

Methodology	OCR
Input	Start frames of two consecutive video shots
Output	Labels of highlighted video shots



Input: video frames of each segmented deliveries
Output: delivery labels that contains the highlights
Algorithm:

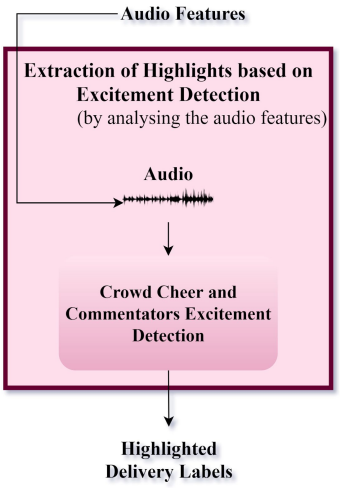
```
1.  i := 1
2.  highlights[total_deliveries]
3.  while i < total_deliveries do :
4.      if isKeyEvent(i,highlights)
5.  end while
6.  return highlights
```

```
isKeyEvent(i,highlights)
1.      if get_score(get_frames(i)) - get_score(get_frames(i+1)) >=4
2.          highlights[i]=1
3.      else  if get_wicket(get_frames(i)) - get_wicket(get_frames(i+1)) >=1
4.          highlights[i]=1
5.      else
6.          highlights[i]=0
7.      end if
```

Extraction of highlights based on excitement detection

This module is used for extracting highlights based on excitement using crowd cheer and commentators excitement from audio. The audio energy of the whole is analysed and computed to get the threshold value then the audio energy of each shots is analysed and computed, checked with the threshold to identify the highlights. Corresponding labels of the shots are considered as the highlights and given to the next module.

Method	Audio energy Analysis
Input	Audio of the input video shots
Output	Labels of highlighted video shots



Input: Recorded sports video sequence and its audio file in .wav format.
Output: Delivery labels containing highlights.
Algorithm:

1. Frames[] :=get_frames(source_video)
2. Paudio := threshold_detection(video)
3. i := 1
4. highlights[total_deliveries]
5. **while** i < total_deliveries do :
6. excitement_detection(Paudio, shot[i])
7. **end** while

threshold_detection(video)

```
1.  for each frame  n do:
2.      x := get_audio_sample(n)
3.      E[n] := short_time_audio_energy(x)
4.  end for
5.  for each frame  n do:
6.          Average_Energy[n] := E[n]
7.  end for
8.  for each frame n do:
9.          NE [n]:= Normalised_Energy(AE[n])
10. end for
11. for each frame n do:
12.                 P_audio   := Average_NE(NE[])
13. end for
    return P_audio
```

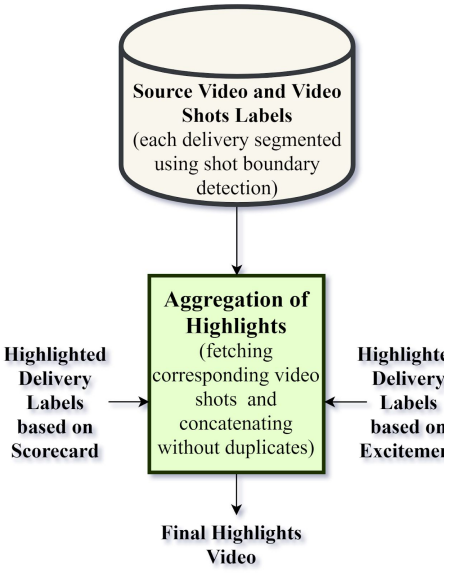
excitement_detection(Paudio ,shot[])

```
1.  i := 1
2.  Highlights[total_deliveries]
3.  while i < total_deliveries do :
4.      for each frame in shots[i]  n do:
5.          x := get_audio_sample(n)
6.          E[n] := short_time_audio_energy(x)
7.      end for
8.      for each frame in shots[i] n do:
9.          Average_Energy[n] := E[n]
10.     end for
11.     for each frame in shots[i] n do:
12.         NE [n]:= Normalised_Energy(AE[n])
13.     end for
14.     for each frame in shots[i] n do:
15.         if  NE(n)>= Average_NE(whole video)
16.                 Highlights[1] :=1
17.         end if
18.     end for
19. end while
20. return highlights
```

Aggregation of Highlights

Highlighted video labels based on scorecard and excitement is combined and corresponding video shots are fetched from the database. These video shots are then concatenated to generate highlights. Highlighted video labels based on scorecard and excitement are combined without duplicates. Corresponding video clips are fetched from the database using the combined labels. These clips are concatenated to form final highlights.

Method	Concatenation of video shots
Input	Highlighted video labels based on Events (scorecard) and Excitement
Output	Final Highlights video



Algorithm:

1) Labels1[] :=HighlightLabels_scorecard,
2) Labels2[] :=HighlightLabels_excitement
3) Result[] :=Label1[] ∪ Label2[]
4) ResultVideo[] := Fetch_video_from_database(Result[])
5) FinalVideo := ConcatenateVideo(ResultVideo)
6) Display(FinalVideo)

IMPLEMENTATION DETAILS

Input: Source video.

Output: Event tagged frames with time.

- ❖ Source video is converted into frames and these frames are converted into gray scale image then into binary image.
- ❖ The binary images are given to OCR for scorecard detection.
- ❖ The OCR recognizes the scorecard in the frames if it all present.
- ❖ Frames with the scorecard are considered as live frames and stored in CSV along with the time.
- ❖ These live frames are given to CNN classifier for event detection.
- ❖ Events like bowling, commentators, interviews are detected.
- ❖ The CNN is trained in order to detect these events. The trained model is loaded to predict the events for the live frames.
- ❖ CSV is updated with the corresponding event tags which is given as output.
- ❖ A shot is considered from one start of the ball to another.
- ❖ With the event tags, shot boundary is detected and segmented.
- ❖ The score of the first frame of each shot is recognized and compared to identify the highlights, shot numbers are given as output.
- ❖ With highlighted shot numbers, corresponding video is fetched and concatenated to form final highlights.

OUTPUT SNAPSHOTS:

Preprocessing:

5_BBL_Trim_preprocessing.csv

No.	time	scorecard present	scorecard absent
1			
2	0s-18s	1	0
3			
4	19s-35s	0	1
5			
6	36s-73s	1	0
7			
8	74s-95s	0	1
9			
10	96s-131s	1	0

Visual Marker Detection:

	time	start of the ball	single_player	commentators	interviews	field_view	crowd	playergathering	others
2	0s	0	0	0	0	0	0	0	0
3	1s	0	0	0	0	0	0	0	0
4	2s	0	0	0	0	0	0	0	0
5	3s	0	0	0	0	0	0	0	0
6	4s	0	0	0	0	0	0	0	0
7	5s	1	0	0	0	0	0	0	0
8	6s	1	0	0	0	0	0	0	0
9	7s	0	0	0	0	0	1	0	0
10	8s	0	0	0	0	0	1	0	0
11	9s	0	0	0	0	0	1	0	0
12	10s	0	0	0	0	0	1	0	0
13	11s	0	0	0	0	1	0	0	0
14	12s	0	0	0	0	1	0	0	0
15	13s	0	0	0	0	1	0	0	0
16	14s	0	0	0	0	1	0	0	0
17	15s	0	0	0	0	1	0	0	0
18	16s	0	0	0	0	1	0	0	0
19	17s	0	0	0	0	1	0	0	0

Shot Boundary Detection and Segmentation:

5_BBL_Trim_BallBoundary.csv

No.	count	start of the ball	end of the ball	score
1				
2	1	5	52	27/1
3				
4	2	52	149	31/1
5				
6	3	149	249	31/2
7				
8	4	249	265	31/3
9				
10	5	265	283	31/3

Extraction of Highlights Based on Scorecard Recognition:

5_BBL_Trim_ScorecardHighlights.csv

No.	count	start of the ball	end of the ball	type
1				
2	1	5	19	Boundary
3				
4	2	52	74	Wicket
5				
6	3	149	167	Wicket

TEST CASES

1. Video with replay/ads
2. Video containing only replay/ads
3. Video without any highlights
4. Video without any ads/replay
5. Video with highlights
6. Video containing only highlights
7. Day Match Video
8. Night Match Video

METRICS FOR EVALUATION

The performance evaluation can be divided into two major categories, namely:

I. Objective evaluation, and II. Subjective evaluation

I. Objective evaluation criterion relies on metrics such as precision, recall, accuracy, false alarm rates, etc. to measure the performance of the system.Following are the some evaluation metrics:

1. **True Positive (TP):** It refers to the positive samples that are correctly labeled by the classifier.
2. **True Negative (TN):** It refers to the negative samples that are correctly labeled by the classifier.
3. **False Positive (FP):** It refers to the negative samples that are incorrectly labeled as positive by the classifier.
4. **False Negative (FN):** It refers to the positive samples that are incorrectly labeled as negative by the classifier.
5. **Precision Rate (PR):** It is a ratio of number of correctly labeled events (or frames), TP, to the total number of events detected.

$$\textbf{\textit{Precision}} = \frac{TP}{TP+FP}$$

6. **Recall Rate (RR):** It is a ratio of true detection rate with respect to the actual events (frames) in the video.

$$\textbf{\textit{Recall}} = \frac{TP}{TP+FN}$$

7. **Error Rate (ER):** It is a ratio of the miss labeled events (both false positives and false negatives) to the total number of events examined.

$$\textbf{\textit{Error}} = \frac{FP+FN}{TP+TN+FP+FN}$$

8. **Accuracy Rate (AR):** It is a ratio of the correctly labeled events (both true positives and true negatives) to the total number of events.

$$\textbf{\textit{Accuracy}} = \frac{TP+TN}{TP+FP+FN+TN}$$

II. Subjective evaluation is based on user feedback score or rating. The user feedback score is generally recorded on a linear scale such as 0: very dissatisfied, 1: somewhat dissatisfied, 2: neutral, 3: satisfied, 4: very satisfied. The subjective nature of this mechanism makes it difficult to define a benchmark as the quality parameters may vary among different users.

Evaluation of preprocessing

Total accuracy= 99% Recall = 99% Precision= 100%

1= Scorecard 2=Replay

		PREDICTED		
		1	**2**	**Accuracy(%)**
ACTUAL	**1**	148	2	99
	2	0	60	100

Evaluation of visual marker detection

Total accuracy= 89% Precision= 62% Recall = 62%

1=Bowling 2= Commentators 3= Interviews
4= Field View 5= Crowd 6= Player gathering
7= Others

		PREDICTED							
		1	**2**	**3**	**4**	**5**	**6**	**7**	**Accuracy (%)**
ACTUAL	**1**	15	0	0	3	0	0	2	83
	2	0	8	1	0	1	0	0	91
	3	0	2	6	0	1	1	0	88
	4	3	0	0	6	0	1	0	88
	5	0	0	2	0	7	1	0	87
	6	0	1	0	0	2	7	0	90
	7	3	0	0	0	0	0	7	91

Evaluation of shot boundary detection and segmentation

Total accuracy=90 % Precision= 63% Recall = 63%

		PREDICTED							
		Ball -1	**Ball -2**	**Ball -3**	**Ball -4**	**Ball -5**	**Ball -6**	**Ball -7**	**Accuracy (%)**
ACTUAL	**Ball 1**	15	0	0	3	0	0	2	83
	Ball 2	0	10	1	0	1	0	0	93
	Ball 3	0	2	6	0	1	1	0	88
	Ball 4	3	0	0	6	0	1	0	88
	Ball 5	0	0	2	0	7	1	0	87
	Ball 6	0	1	0	0	2	7	0	90
	Ball 7	3	0	0	0	0	0	7	91

Evaluation of extraction of highlights based on scorecard recognition

Total accuracy=89 % Precision= 62% Recall = 62%

Here, each number corresponds to the video shots number in highlight

		PREDICTED					
		1	**2**	**3**	**4**	**5**	**Accuracy (%)**
ACTUAL	**1**	15	0	0	3	0	83
	2	0	8	1	0	1	91
	3	0	2	6	0	1	88
	4	3	0	0	6	0	88
	5	0	0	2	0	7	87

REFERENCES

[1] Michele Merler, Khoi-Nguyen C. Mac, Dhiraj Joshi, Quoc-Bao Nguyen Stephen Hammer, John Kent, Jinjun Xiong, Minh N. Do, John R. Smith, Rogerio S. Feris., **"Automatic Curation of Sports Highlights using Multimodal Excitement Features"**, in *IEEE Transactions on Multimedia.* doi: 10.1109/TMM.2018.2876046

[2] A. Javed, K. B. Bajwa, H. Malik and A. Irtaza, **"An Efficient Framework for Automatic Highlights Generation from Sports Videos,"** in *IEEE Signal Processing Letters,* vol. 23, no. 7, pp. 954-958, July 2016. doi: 10.1109/LSP.2016.2573042

[3] M. H. Kolekar and S. Sengupta, **"Bayesian Network-Based Customized Highlight Generation for Broadcast Soccer Videos,"** in *IEEE Transactions on Broadcasting,* vol. 61, no. 2, pp. 195-209, June 2015. doi: 10.1109/TBC.2015.2424011

[4] Shukla, Pushkar & Sadana, Hemant & Bansal, Apaar & Verma, Deepak & Elmadjian, Carlos & Raman, Balasubramanian & Turk, Matthew, **"Automatic Cricket Highlight Generation Using Event-Driven and Excitement-Based Features"**, 2018 1881-18818. 10.1109/CVPRW.2018.00233.

[5] Chih-Chieh Cheng and Chiou-Ting Hsu, **"Fusion of audio and motion information on HMM-based highlight extraction for baseball games,"** in *IEEE Transactions on Multimedia*, vol. 8, no. 3, pp. 585-599, June 2006. doi: 10.1109/TMM.2006.870726

[6] M. H. Kolekar, K. Palaniappan and S. Sengupta, **"Semantic Event Detection and Classification in Cricket Video Sequence,"** *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, Bhubaneswar, 2008, pp. 382-389. doi: 10.1109/ICVGIP.2008.102.

[7] Kolekar, Maheshkumar H., and S. Sengupta. **"Caption Content Analysis Based Automated Cricket Highlight Generation."** National Communications Conference (NCC), Mumbai. 2008.

[8] M. E. Anjum, S. F. Ali, M. T. Hassan and M. Adnan, **"Video summarization: Sports highlights generation,"** *INMIC*, Lahore, 2013, pp. 142-147. doi:10.1109/INMIC.2013.6731340.

[9] Hasan, Taufiq, et al. **"A multi-modal highlight extraction scheme for sports videos using an information-theoretic excitability measure."** Acoustics, Speech and Signal Processing (ICASSP), *2012 IEEE International Conference* on. IEEE, 2012.

[10] D. Tjondronegoro, Y.-P.P. Chen,, B. Pham,, **"Sports Video Summarization Using Highlights and Play-Breaks"**, Proc. *ACM SIGMM Int'l Workshop Multimedia Information Retrieval,* pp. 201-208, 2003.

Works Cited

Anjum, Muhammad Ehsan, et al. “Video Summarization: Sports Highlights Generation.” *Inmic*, 2013, doi:10.1109/inmic.2013.6731340.

Cheng, Chih-Chieh, and Chiou-Ting Hsu. “Fusion of Audio and Motion Information on HMM-Based Highlight Extraction for Baseball Games.” *IEEE Transactions on Multimedia*, vol. 8, no. 3, 2006, pp. 585–599., doi:10.1109/tmm.2006.870726.

Hasan, Taufiq, et al. “A Multi-Modal Highlight Extraction Scheme for Sports Videos Using an Information-Theoretic Excitability Measure.” *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012,

doi:10.1109/icassp.2012.6288394.

Javed, Ali, et al. “An Efficient Framework for Automatic Highlights Generation from Sports Videos.” *IEEE Signal Processing Letters*, vol. 23, no. 7, 2016, pp. 954–958., doi:10.1109/lsp.2016.2573042.

Kolekar, Maheshkumar H., and Somnath Sengupta. “Bayesian Network-Based Customized Highlight Generation for Broadcast Soccer Videos.” *IEEE Transactions on Broadcasting*, vol. 61, no. 2, 2015, pp. 195–209., doi:10.1109/tbc.2015.2424011.

Kolekar, M.h., et al. “Semantic Event Detection and Classification in Cricket Video Sequence.” *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, 2008, doi:10.1109/icvgip.2008.102.

Merler, Michele, et al. “Automatic Curation of Golf Highlights Using Multimodal Excitement Features.” *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, doi:10.1109/cvprw.2017.14.

Shukla, Pushkar, et al. “Automatic Cricket Highlight Generation Using Event-Driven and Excitement-Based Features.” *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, doi:10.1109/cvprw.2018.00233.

Tjondronegoro, Dian, et al. “Sports Video Summarization Using Highlights and Play-Breaks.” *Proceedings of the 5th ACM SIGMM International Workshop on Multimedia Information Retrieval - MIR '03*, 2003, doi:10.1145/973264.973296.