International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 06 | Jun 2024 www.irjet.net p-ISSN: 2395-0072

Automatic Key-Event Extraction from Sports Videos Using Scoreboard Detection

[1]SINDHUJA K, [2]EYAMINI C, [3]BRINDHA S

[1]Assistant Professor, Department of Computer Science and Engineering, EASA College of Engineering and Technology, Coimbatore, Tamil Nadu, India
[2]Assistant Professor, Department of Artificial Intelligence and Data Science, Kathir College of Engineering, Coimbatore, Tamil Nadu, India
[3]Assistant Professor, Department of Computer Science and Engineering, EASA College of Engineering and Technology, Coimbatore, Tamil Nadu, India

---

Abstract: *This paper introduces an efficient method for extracting key events from sports videos by leveraging scoreboard detection. The approach involves training a supervised learning-based object detection algorithm (YOLO) on a dataset of 1200 images. Once the scoreboard is detected in video frames, it is cropped out, and image processing techniques are applied to reduce noise. An Optical Character Recognizer (OCR) extracts the score, and a rule-based algorithm generates precise timestamps for critical moments during the game. The proposed method achieves an average F1 Score of 0.979 across different sports, making it valuable for sports analysis. This implementation is in Python 3.7.*

Index Terms - Automatic Key-Event Extraction, Sports Videos, Scoreboard Detection, YOLO (You Only Look Once), Image Processing, Optical Character Recognizer (OCR), Timestamp Generation.

1.INTRODUCTION

The Entertainment Industry and Sports Videos Sports play a significant role in the entertainment industry, attracting a vast audience worldwide. With the increasing accessibility of the internet, sports viewership has surged. However, many individuals struggle to find time to watch full-length matches due to their busy lives. As a result, there is a growing preference for shorter, summarized versions of sports events. Manual summarization or highlight generation is a labor-intensive process that requires professional video editing tools. Unfortunately, this approach imposes limitations on the amount of video footage that can be effectively summarized within a given timeframe.

2. LITERATURE

Some of the previous work in this area includes the use of ORB (Oriented Fast, Rotated Brief) to identify BRS (bowler run sequence) in cricket and classify them as important [1]. Pushkar Shukla et al. In [3], they used the intensity of the operator's gestures to determine the occurrence of events. A. Javed et al. wicket). Extracting features from football videos for content using 3D CNN (3D convolutional neural network) and LSTM (long-term memory) [5] and [6]. Shingrakhia et al. (network). J.Yu et al. Although these are new techniques, they all focus on a single aspect for the subject and are difficult to translate to other movements. Others, such as P. Kathirvel and others, have attempted to generalize across different sports. This works for games that don't require a whistle or for background noise etc. It fails when the whistle cannot be heard due to the neural network works and uses it to identify important events in the game called Kendo (Japanese fencing). This includes cricket, football etc. It doesn't work very well in multiplayer games like. In this paper, we propose a computationally cheap method to extract outcome-based values from a scoreboard about a game. Call YOLO and train it to determine the desired score of the game. We then run a sample of images taken from the movie at a frame rate of one frame per second and run OCR on them to keep track of the scores. If the score is changed by some forward increment (for example, in a baseball match), the increment of the variable score will be set to 4/6 and the increment of the wicket difference will be set to 1 because the difference between the wicket1 indicates that it is an event and therefore should occur in the context of the match. In football, the delta will appear during the match and the increment will be closed correctly. Our model is very simple, only need 250-300 points of footage to train and anyone can use it for any sports video using the method with knowledge of sports to adjust the time increment. The remainder of this article is organized as follows. Section 3 describes YOLO and why we chose YOLO, Section 4 presents the methodology, Section 5 presents the results and performance of our model, and finally Section 6 presents conclusions and limitations/future work.

2.1 YOLO (You Only Look Once)

2.1.1 COMPARISON OF OBJECT DETECTION ALGORITHMS

There are many target detection algorithms such as R-CNN [12], Faster R-CNN [13], and YOLO. In this section, we will discuss the reasons why we prefer YOLO. This is done by showing other algorithms, their limitations, and how YOLO addresses/overcomes these limitations. Region based convolution Neural Network (R-CNN): The aim is to divide

the image into 2000 small areas and then find the objects in each area and also add the distances to improve the reality. Limitations: It takes a lot of time to rank 2000 sites. It is not applicable for instant detection as each image requires approximately 47 seconds processing (classification). Faster R-CNN: The network itself learns regional recommendations. It is faster than R-CNN and R-CNN [14], but the algorithm is still not suitable in terms of performance. It can be seen that almost every algorithm relies on three tasks to achieve the detection goal: 1) Use the Regional Recommendation method to create potential bounding boxes in the image. 2) Run the classifier on these boxes. 3). This is where YOLO comes into play. Predict bounding boxes and probability using a convolutional neural network. Predictions with the threshold (maximum IOU) are considered detection and the rest are discarded. It is currently the fastest algorithm available, with a speed of approximately 45 frames per second. Limitation: Due to the limited space of the algorithm, the algorithm cannot detect small objects.

2.1.2 YOLO (YOU ONLY LOOK ONCE) ALGORITHM AND ITS KEY COMPONENTS

The algorithm divides the input image into an S × S grid. If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object. Each grid cell predicts a fixed number of boundary boxes and confidence scores for those boxes to locate the object. Training data is a bounding box along with the class label(s). This is referred to as 'ground truth box', Grid-Based Detection: 1) YOLO divides the input image into an S × S grid. 2) Each grid cell is responsible for detecting objects that fall within its boundaries. 3) If the center of an object falls into a grid cell, that cell is tasked with detecting that object. Bounding Box Prediction: 1) For each grid cell, YOLO predicts a fixed number of bounding boxes (rectangular regions) and their associated confidence scores. 2) These bounding boxes represent potential object locations. 3) The predicted bounding boxes include information about the midpoint (bx, by), height (bh), and width (bw). Class Prediction: 1) YOLO also predicts class probabilities for each detected box. 2) It assigns a class label (e.g., "person," "car," "soccer ball") to the detected object. 3) The class probabilities are represented by the vector cn, where n corresponds to the number of object classes the model can detect. Linear Activation Function: 1) YOLO uses a linear activation function for the final layer. 2) For other layers, it employs a leaky rectified linear activation function. Loss Function: 1) YOLO calculates the loss using sum-squared error between predictions and ground truth. 2) The loss function includes three components: **Classification Loss:** Squared error of class conditional probabilities for each class. **Localization Loss:** Measures errors in predicted boundary box locations and sizes (considering only the box responsible for detecting the object). **Confidence Loss:** Associated with the confidence score for each bounding box predictor. The final loss combines these components.

3. RESEARCH METHODOLOGY

3.1 Training the YOLO Model

The model was trained using a list of different scores consisting of approximately 250 to 300 images (a total of approximately 1300 images were used for one category) [16]. Of the 300 images in a group, approximately 200 to 240 were used as training data and the remaining images were used as testing/validation data. Percentage of data is used for training and 20% is used for validation. The model was trained approximately 11 times with size set to 4 and IOU set to 0.5 (out of 11 times, each model dropped below 1). A total of 3 models are trained (using pre-trained models provided by Image AI and retraining them), 1 for each game (cricket, football, kabaddi). 1. The score format using 250 images can produce very good results.

3.2 Feeding the Images to the Model

Using OpenCV we can extract one frame per second from a video sequence (e.g. 1 in 30 frames per second for 30FPS video) and feed it into the model. Whether the sample output image contains scores If so, return the combination of the image's scores. The rate of the feed image is kept at 1 frame per second of the video sequence so that the error between the output time of the event and the correct timestamp of the event is more than 1 second. Although a lower value reduces job complexity and improves performance, it still causes the most errors; for example a value of 0.2 frames per second in a video interval or an interval of 1 image per second > 5 seconds will result in a maximum error of 5 seconds which is unacceptable.

3.3 Reading the Score from the Scoreboard

If the model finds the score in the image, the score will be deducted from the image. Then, the following image operations will be performed on the image obtained from the score: use threshold to convert the image to binary, find the contours of the image and define the ROI (image region) around each contour as separate contours, crop the areas from the image to create separate images, each containing 1 number or symbol. All these regions are fed to the trained OCR [17], which returns the score in the current frame. There are many OCRs on the internet, but none of them give good results in reading the scores because after cropping the score image becomes smaller and hence the resolution issues decrease. Therefore, we trained a convolutional neural network (including the hidden layer) on the 5 code sets used in this project. This means that the classification accuracy of OCR goes from zero to almost 100%. The standard model has 1 CNN layer, the filter size is 40, the hidden layer has 128 neurons, the kernel size is 3x3, the model is trained using adam optimizer and sparse classification cross entropy loss function. The model has 14 neurons in the output layer, corresponding to the 14 groups of characters that it can recognize, which are numbers (0, 1,

2, 3, 4, 5, 6, 7, 8, 9) and the training scores. There are the following special characters: '|', 'P', '-' and ':'(The characters '|', '-' and ':' are reserved in the generics and the symbol "P" means: a power play in cricket.

3.4 Steps to Minimize False Positive in OCR

1. Before processing the video, 10 frames (each taken 30 seconds apart) are extracted from the video and fed to YOLO. The output locations of these 10 frames are averaged to get a tighter bound on the scoreboard.
2. A threshold on the minimum size of individual contours is used to eliminate granular noise.
3. Regular Expression pattern matching is used to eliminate random scores such as 12-3-0.

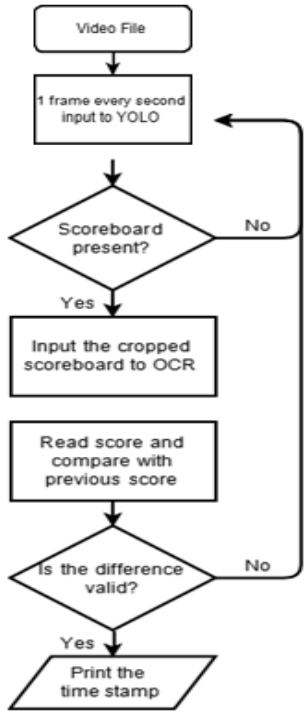

Figure 1: Flow Chart of the Algorithm

3.5 RegEx

A RegEx, or Regular Expression, is a special sequence of characters that helps you match or find other strings or sets of strings, using a specialized syntax held in a pattern [18]. RegEx can be used to check if a string contains the specified pattern of characters. Regular expressions are used extensively for pattern matching everywhere. For example in online forms, a regular expression is generally used to verify that the email id entered by the user is a valid email id like so: r'([a-zA-Z0-9.])+@(gmail|outlook|apple|live|hotmail|yahoo)(.com)'. This will match a string containing [a-zA-Z0-9.] - any combination of characters a to z (upper and lower case), digits, periods and underscores followed by the @ character followed by (gmail|outlook|apple|live|hotmail|yahoo) - the email domains Gmail, Outlook, Apple, Live, Hotmail, yahoo followed by .com. If a match of the pattern is found in the string input by the user, then it is considered a valid email id. RegEx is also used to check the strength and validity of a password, for example, r'[a-zA-Z0-9@# &]{8,}' can be used to only accept a password that is at least 8 characters long and contains only a-z (upper and lower case) digits and special characters @,#,-, and &. Similarly, a phone number can be verified using r'(\+91)?(\d{10}|\d{8})'. This will match a string that contains only digits, is either 10 or 8 characters long, and has an optional +91 at the front.

3.6 Algorithm to Detect an Event

The score output by OCR will be compared with the score of the previous frame and the difference between the scores will be used to determine whether an event occurred in the current frame as some prerequisites. For example - If the difference in a sports match is 6, 4, 3, 5 or the wicket difference is 1, it is considered an event and the duration of the Custom frame should be displayed along with the type of event that occurred. Using these time logs, landmarks can be extracted using static numbers before and after the event is detected (for example, 7 seconds before the event and 7 seconds after the event).

4. RESULTS AND DISCUSSION

4.1 Performance of Trained Scoreboard Detection Models

All YOLO models (3 models, 1 model per game group) were tested on images of their own level games using an IOU value of 0.5. These images are not part of the training materials and are in different resolutions (between 360p and 1080p) to ensure standard image quality. The results are shown in Table 1.

Model	Input Images	Images with a detection	Best mAP
Kabaddi	647	639	1.00
Soccer	639	600	0.93
Cricket	341	341	0.98

Table 1: Performance of Trained Detection Models

4.2 Performance on Video Footage

The plan was tested with a total of 6 hours 35 minutes of kabaddi, football and cricket video clips. The performance metrics used here are Precision, Recall, and F1 Score, and the IOU value is set to 0.5.

Some terms related to precision, recall, and accuracy (the following definitions have been derived from [19]):

True positives – These are the correctly predicted positive values which means that the actual class (ground truth) is 'scoreboard' and the predicted class is also 'scoreboard'.

True negatives - These are the correctly predicted negative values which means that the actual class (ground truth) is 'not scoreboard' and predicted class is also 'not scoreboard'.

False positives - These are the wrongly predicted positive values which means that the actual class (ground truth) is 'no scoreboard' and predicted class is 'scoreboard'.

False negatives - These are the wrongly predicted negative values which means that the actual class (ground truth) is 'scoreboard' but predicted class in 'not scoreboard'.table 1 table type styles.

Class	**Predicted: 'scoreboard'**	**Predicted: 'not scoreboard'**
Ground Truth: 'scoreboard'	True Positives	False Negatives
Ground Truth: 'not scoreboard'	False Positives	True Negatives

Table 2: Confusion Matrix

Precision is defined as the ratio of correctly predicted observations to the total predicted positive observations.

$$\text{Precision} = \frac{TP}{TP+FP} \quad (5)$$

Recall is defined as the ratio of correctly predicted positive observations to the total observations in the actualclass.

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

F1 Score is defined as the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account.

$$\text{F1 Score} = \frac{2 * (Precision * Recall)}{Precision + Recall}$$

Since the model analyses 1 frame from 1 second of the video sequence, the detected events are within 1 second of their actual occurrence. Figures 2 through 11 show the working of the model by depicting images at different stages of the algorithm.

Game	Actual Events	Detected Events	TP	FP	FN	Precision	Recall	F1 Score
Kabaddi	75	78	75	3	0	0.961	1	0.980
Soccer	11	11	11	0	0	1	1	1
Cricket	108	105	102	3	6	0.971	0.944	0.957

Table 3: Performance on Video Footage

Figure 2: A Frame from the Video

Figure 3: Output of YOLO

Figure 4: Cropped Scoreboard

Figure 5: Input to the OCR

```
0:9 ~ Four
0:13 ~ 3 or 5 runs
0:16 ~ Wicket
0:38 ~ Six
0:45 ~ Four
0:56 ~ Wicket
0:56 ~ Six
1:1 ~ Four
```

Figure 6: Output of the Algorithm

5. Conclusion, Limitations and Future Work

This paper presents a deep learning model to record sports videos and display the timing of important events. It uses YOLO to analyze the score and then some python code to detect the result. The model was implemented using Python 3.7 and TensorFlow 1.13 on a computer running Windows 10 with a 1.90 GHz, Core i5-4460T CPU, 4 GB RAM. Other software used for this model includes OpenCV, Keras API, and Python's ImageAI module. The average (all groups) F1 score of the output is 0.979. The model can be trained and sent to other groups (for example, it can be used to write

instructions for videos). Below we describe the limitations of the current implementation and describe future work:

Figure 7: A Frame from the Video

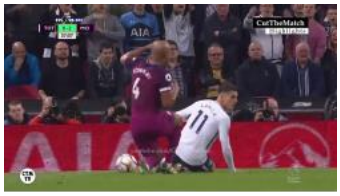

Figure 8: Output of YOLO

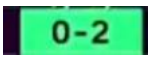

Figure 9: Cropped Scoreboard

Figure 10: Input to the OCR

```
4:57 ~ Away team scored
6:23 ~ Away team scored
8:13 ~ Away team scored

Goals Scored by the Home Team: 0
Goals Scored by the Away Team: 3
```

Figure 11: Output of the Algorithm

- A need for custom OCR emerged as the existing OCRs could not detect the text or were giving false output. The OCR used in the project works on very specific kinds of texts, we would like to make it more generalized as part of a future project.
- The algorithm currently works for mp4 videos only, we plan to extend the algorithm to work on differentvideo formats.
- Because of limitations of hardware, video clips extraction was not possible and only the timestamps of key events are the output. In the future, we plan to extract the highlight clips.

We also plan to generalize the scoreboard detection model to detect any kind of scoreboard so as to mitigate the need to train the model on new scoreboard classes. This can be achieved by training the model on a very large image database.

Here we conclude our paper and express our heartfelt gratitude towards Professor Hansa Shingrakhia, who mentored us throughout our journey.

References

[1] D. Ringis and A. Pooransingh, "Automated highlight generation from cricket broadcasts using orb," in 2015 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM),pp. 58–63, 2015.

[2] P. Shukla, H. Sadana, A. Bansal, D. Verma, C. Elmadjian, B. Raman, and M. Turk, "Automatic cricket highlight generation using event-driven and excitement-based features," in 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1881–18818, 2018.

[3] Hari R. and Wilscy M., "Event detection in cricket videos using intensity projection profile of umpire gestures," in 2014 Annual IEEE India Conference (INDICON), pp. 1–6, 2014.

[4] A. Javed, A. Irtaza, Y. Khaliq, H. Malik, and M. Mahmood, "Replay and key-events detection for sports video summarization using confined elliptical local ternary patterns and extreme learning machine," Ap-plied Intelligence, 02 2019.

[5] R. Agyeman, R. Muhammad, and G. S. Choi, "Soccer video summarization using deep learning," in 2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 270–273, 2019.

[6] M. Z. Khan, S. Saleem, M. A. Hassan, and M. Usman Ghanni Khan, "Learning deep c3d features for soccer video event detection," in 2018 14th International Conference on Emerging Technologies (ICET),pp. 1–6, 2018.

[7] H. Shingrakhia and H. Patel, "Emperor penguin optimized event recognition and summarization for cricket highlight generation," Multimedia Systems, vol. 26, no. 6, pp. 745–759, 2020.

[8] H. Shingrakhia and H. Patel, "Sgrnn-am and hrf-dbn: a hybrid machine learning model for cricket video summarization," The Visual Computer, 04 2021.

[9] Y. Junqing, L. Aiping, and H. Yangliu, "Soccer video event detection based on deep learning," in Multi-Media Modeling, (Cham), pp. 377–389, Springer International Publishing, 2019.

[10] P. Kathirvel, M. Manikandan, and K. Soman, "Automated referee whistle sound detection for extraction of highlights from sports video," International Journal of Computer Applications, vol. 12, 12 2011.

[11] A. Tejero-de-Pablos, Y. Nakashima, T. Sato, N. Yokoya, M. Linna, and E. Rahtu, "Summarization of user-generated sports video by using deep action recognition features," IEEE Transactions on Multimedia, vol. 20, no. 8, pp. 2000–2011, 2018.

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," CoRR, vol. abs/1311.2524, 2013.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in Advances in neural information processing systems, pp. 91–99, 2015.

[14] R. Girshick, "Fast r-cnn," in Proceedings of the IEEE international conference on computer vision,pp. 1440–1448, 2015

[15] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv, 2018.

[16] "How to train a yolo model." https://medium.com/deepquestai/train-object-detection-ai-with-6-lines-of-code-6d087063f6ff.

[17] "Digit recognition using cnn." https://towardsdatascience.com/image-classification-in-10-minutes-with-mnist-dataset-54c35b77a38d.

[18] "Accuracy, precision, recall and f1 score: Interpretation of performance measures." https://blog.exsilio.com/all/accuracy-precision-recall-f1-score-interpretation-of-performance-measures/.

[19] "Google colaboratory." https://colab.research.google.com.

[20] C. Guntuboina, A.Porwal, P.Jainand, H.Shingrakhia, "Video summarization for multiple sports using deep learning,"2022 In:Proceedings of the international e-conference on intelligent systems and signal processing, vol.1370, pp643–656. doi:10.1007/978-981-16-2123-9_50.