Signal, Image and Video Processing (2025) 19:232
https://doi.org/10.1007/s11760-024-03805-x

ORIGINAL PAPER

Capturing captivating moments: a multi-model approach for identifying baseball strikeout highlights

Qiaoyun Zhang[1] · Chih-Yung Chang[2] · Cuijuan Shang[1] · Hsiang-Chuan Chang[3] · Diptendu Sinha Roy[4]

Received: 22 September 2023 / Revised: 19 December 2024 / Accepted: 27 December 2024

Abstract
With the extensive popularity of baseball, fans are eager to relive exciting moments such as strikeouts, catching, and home runs. However, manually extracting these highlights from long videos is time-consuming and labor-intensive. To address this, this paper introduces a mechanism called CCM (capturing captivating moments), which aims to effectively identify baseball strikeout highlights. Initially, the proposed CCM employs a coarse-grain policy that involves two key components. Firstly, it utilizes You Only Look Once (YOLO) to detect the change of out-indicator in video frames. Secondly, it employs long short-term memory to analyze the skeleton features of athletes, aiming to extract the draft segments as the candidates that contain the strikeout. Then a fine-grain policy integrates YOLOv5, bidirectional encoder representations from transformers, and 3D convolutional neural networks to accurately identify the strikeout highlights. By combining heterogeneous features and multi-model integration, the proposed CCM ensures robust and precise identification of captivating strikeout moments in baseball videos. The simulation results demonstrate that the proposed CCM outperforms the existing mechanisms in terms of accuracy, recall, precision, and F1-score.

Keywords Action recognition · Object detection · Heterogeneous features · Multi-model integration

1 Introduction

Baseball is a sport that garners worldwide attention. Many fans have a strong desire to relive exciting moments such as strikeouts, catching, or home runs. However, the task of extracting these highlights from lengthy baseball videos is both time-consuming and labor-intensive. Exploring how technology can be utilized to identify custom short clips from a long video is an important and commercially valuable research endeavor. It involves applying Artificial Intelligence (AI) technology [1–4] to analyze and understand the actions depicted in baseball videos.

This paper aims to utilize AI to extract the highlights from baseball videos. By accurately detecting objects and identifying the behaviors of the athletes in the videos, the proposed CCM can automatically identify strikeout highlights based on the distinctive features of the baseball videos. This approach revolutionizes the video editing process, making it more efficient, precise, and cost-effective in terms of time and manpower.

Previous research [5–10] on object detection has witnessed significant advancements with the success of deep convolutional neural networks (CNN). The mainstream anchor-based object detectors can be categorized into two types: two-stage object detectors, such as R-CNN (Regions with CNN features) [5] and Fast R-CNN [6], and one-stage object detectors, including Single-Shot multibox Detector

✉ Chih-Yung Chang
cychang@mail.tku.edu.tw

Qiaoyun Zhang
zqyun@chzu.edu.cn

Cuijuan Shang
shangcuijuan@chzu.edu.cn

Hsiang-Chuan Chang
149190@o365.tku.edu.tw

Diptendu Sinha Roy
diptendu.sr@nitm.ac.in

1 School of Computer and Information Engineering, Chuzhou University, Chuzhou, China

2 Department of Computer Science and Information Engineering, Tamkang University, New Taipei, Taiwan

3 Department of Transportation Management, Tamkang University, New Taipei, Taiwan

4 Department of Computer Science and Engineering, National Institute of Technology, Shillong, India

Published online: 23 January 2025

(SSD) [7] and YOLO [8–10]. Among these, the YOLO has received extensive attention due to its effectiveness and real-time performance. However, creating highlights of baseball strikeouts requires analyzing multiple consecutive images to make decisions about which segment to extract. Therefore, the techniques that solely process individual images are more challenging to apply in addressing the topic discussed in this research paper.

Previous research on human action recognition has primarily focused on two types of features: Red Green Blue (RGB) and skeleton features. RGB-based methods [11, 12] aimed to extract spatial and temporal features from RGB frames and/or temporal optical flow to recognize behaviors. While these methods have shown promising results, they still face challenges such as background clutter, illumination changes, and appearance variations. In contrast, skeleton features represent the body structure through a series of coordinate positions of key joints. These features exhibit viewpoint and appearance invariance, reducing discrepancies within the same class when compared to RGB features. Additionally, skeleton features exclusively capture high-level information that describes human movements. Yao et al. [13] demonstrated that skeleton-based features outperformed appearance-based features when using the same classifier on the same dataset.

This paper proposes a mechanism, called CCM, which aims to effectively identify strikeout highlights in baseball videos. The proposed CCM utilizes YOLO to detect various objects in the video frames. Additionally, it analyzes the skeleton features and subtitles associated with the video to automatically extract the most significant features of the strikeout highlights. Initially, the proposed CCM provides a coarse-grain policy that utilizes YOLOv5 [14, 15] and LSTM [16, 17] to obtain the candidate video. The YOLOv5 model is employed to detect the various out-indicator, facilitating the determination of the end time of candidate videos. Furthermore, both YOLOv5 and LSTM models are employed to detect the 'person' and analyze the time-sequence skeletons of each 'person' object, respectively, enabling the identification of the start time of candidate videos. To accurately identify which candidate video is the strikeout highlight, the proposed CCM further adopts a fine-grain policy that integrates three distinct approaches: YOLOv5, BERT [18], and 3DCNN [19] to select the strikeout highlight from the candidates. YOLOv5 is employed to detect the presence of the letter 'K', commonly indicating a strikeout. BERT analyzes the subtitles in the candidate video to determine their relevance to 'strikeout.' Additionally, 3DCNN focuses on learning short-term spatiotemporal features associated with 'strikeout.' These three models are integrated to maximize recognition accuracy. The main contributions of the paper are summarized as follows:

1) ***Efficient candidate identification through coarse-grained design.*** The proposed CCM adopts a coarse-grained design to efficiently detect key features and identify candidate videos. One feature is the out-indicator light. The YOLOv5 is applied to detect changes in the out-indicator lights, allowing for the determination of the end time of candidate videos. Additionally, both YOLOv5 and LSTM models are employed to analyze the skeletal data of individuals in the video, facilitating the identification of candidate videos' starting times. This coarse-grained design speeds up the identification process and enhances efficiency.
2) ***Fine-grained design for accurate strikeout highlights identification.*** The proposed CCM introduces three distinct models, including YOLOv5, BERT, and 3DCNN, to precisely identify whether a candidate video is a strikeout highlight. YOLOv5 focuses on detecting the presence of the letter 'K' in the video while the BERT model analyzes the subtitles to check if they are related to 'strikeout'. In addition, the 3DCNN learns short-term spatiotemporal features associated with the 'strikeout'. These three models are integrated to achieve optimal recognition accuracy.
3) ***Combining heterogeneous features and multi-model to completely explore the features from baseball videos.*** The proposed CCM combines the capabilities of YOLO, 3DCNN, and BERT to effectively identify strikeout highlights by leveraging heterogeneous features for a given video. By synergistically combining the outputs of these three approaches, the CCM comprehensively evaluates and makes a final decision regarding the identification of strikeout highlights, ensuring a robust and accurate outcome.

The remainder of the paper is organized as follows. Section II discusses and compares previous relevant studies. Section III describes the detailed notations, assumptions, problem descriptions, and objectives. Section IV presents the detailed proposed CCM mechanism. Section V provides the simulation and performance evaluation. The summary and future work are discussed in section VI.

2 Related work

In this section, this paper reviews the existing work that closely relates to the proposed CCM mechanism. These studies were primarily partitioned into two categories: object detection and human action recognition.

2.1 Object detection

Object detection is a pivotal task in computer vision. Methods in object detection can be categorized into two main groups: two-stage [5, 6] and one-stage approaches [7–10]. R-CNN proposed by Girshick et al. [5] was one of the earliest methods to utilize CNN for object detection. Building on R-CNN, Girshick [6] proposed a Fast R-CNN incorporating innovations to enhance training and testing speeds while improving detection accuracy.

In contrast to the two-stage approach, the one-stage approach predicted and classified candidate boxes directly at multiple image locations, eliminating the need for pre-generated candidate boxes. The most representative examples of single-stage approaches are SSD and YOLO. Zhao et al. [7] proposed SSD, which detected objects of different scales at different layers of the network. On the other hand, Redmon et al. [8] proposed YOLO, which treated object detection as a regression problem, associating each box with a probability and spatially separating bounding boxes. YOLO employed a single neural network to directly predict bounding boxes and class probabilities from full images in one step, optimizing the pipeline directly based on detection performance. However, when it comes to identifying strikeout highlights in baseball videos, these approaches that solely process individual images face significant challenges. Extracting strikeout highlights requires analyzing multiple consecutive images to make informed decisions about which segments to extract. This complex temporal analysis is not adequately addressed by the approaches.

To overcome this limitation and improve detection accuracy, this paper incorporates improved YOLOv5s and LSTM models. The improved YOLOv5s algorithm reduces the number of categories compared to the original version, while the LSTM model enables the analysis of temporal sequences. By combining these techniques, the paper aims to achieve more accurate and robust detection results.

2.2 Human action recognition

3DCNN is mainly used for video action recognition in videos [20–23]. Liu et al. [21] proposed a temporal convolutional 3D network (T-C3D), which employed RGB frames to learn hierarchical multi-granularity video action representations. Khong [22] employs two-stream 3DCNNs using RGB and optical flow computed from RGB stream as inputs. Zhao et al. [23] combined the 3DCNN and support vector machine (SVM) for video action recognition, introducing an adaptive keyframe extraction strategy based on scene change speed and RGB histogram analysis. Extracted keyframes were used to train a 3DCNN for feature extraction, subsequently fed into an SVM for action recognition. However, the effectiveness of the keyframe extraction strategy might be influenced by its sensitivity to scene changes. Additionally, the 3DCNN appeared to have limitations in representing long variations, thus potentially impacting its ability to capture the strikeout highlight in baseball videos. To capture the dynamic dependencies in sequential data, researchers turned to RNNs and LSTMs. Several methods [24–27] adopted these models.

to identify human action identification using skeleton features. Du et al. [24] proposed an end-to-end hierarchical RNN approach, dividing the human skeleton into five body parts. These parts were separately processed by multiple bidirectional RNNs, and their output representations were hierarchically fused to generate high-level action representations. This approach might simplify the learning process. However, it might overlook intricate joint interactions crucial for understanding actions.

Zhang et al. [25] introduced a multi-stream LSTM framework with enhanced score fusion for skeleton-based behavior recognition. This approach aimed to extract classification knowledge from various geometric feature streams effectively. However, the behaviors recognized by these approaches were relatively regular. In baseball competitions, each pitching may exhibit unique image patterns. The proposed CCM combines YOLO and LSTM models to enhance pitching identification accuracy. By leveraging the object detection capabilities of YOLO and the sequential modeling abilities of LSTM, the system can accurately identify and classify the diverse pitching actions observed in baseball competitions.

Graph structures have gained prominence for analyzing data. Several graph neural networks and Graph Convolutional Networks (GCN)-based methods [28–35] for human action recognition treat skeleton data as graphs. Yan et al. [28] proposed the spatiotemporal graph convolutional network (ST-GCN), which constructed a spatiotemporal graph to encode the skeleton sequences and stacked a set of ST-GCN to extract features and make predictions. However, the ST-GCN relied on the predefined skeleton graph, which might limit the performance of the model in action recognition. To better catch the implicit joint correlations, Shi et al. [31] proposed a multi-stream attention-enhanced adaptive graph convolutional network (MS-AAGCN) for skeleton-based action recognition. The graph topology in MS-AAGCN could be either uniformly or individually learned based on the input data. Although MS-AAGCN achieved considerable performance, the increased computational cost associated with the multi-stream structure poses a challenge. To address this, recent work in multimodal fusion by Yang et al. [38] [39] proposed a discrete multi-view anchor-graph clustering approach, which solved the discrete multi-view graph-cutting problem to generate a discrete cluster indicator matrix.

Table 1 The comparisons of the proposed CCM and the related work

Method	RGB	Skeleton	Subtitles	Pattern	Temporal	Spatial	Models
Liu et al. [21]	✓	✗	✗	Static	✓	✗	3DCNN
Khong et al. [22]	✓	✗	✗	Static	✓	✓	Two-stream 3DCNNs
Zhao et al. [23]	✓	✗	✗	Static	✓	✓	3DCNN + SVM
Du et al. [24]	✗	✓	✗	Static	✓	✓	RNN
Zhang et al. [25]	✗	✓	✗	Static	✓	✓	Multi-stream LSTM
Zhang et al. [26]	✗	✓	✗	Static	✓	✓	LSTM + CNN
Tang et al. [27]	✗	✓	✗	Static	✓	✓	self-supervised learning
Yan et al. [28]	✗	✓	✗	Static	✓	✓	ST-GCN
Zhang et al. [30]	✗	✓	✗	Static	✓	✓	GCN
Shi et al. [31]	✗	✓	✗	Static	✓	✓	MS-AAGCN
Yang et al. [32]	✗	✓	✗	Static	✓	✓	Feedback GCN
Yang et al. [38]	✓	✓	✗	Dynamic	✓	✓	Fast multi-view anchor graph clustering
Yang et al. [39]	✓	✓	✗	Dynamic	✓	✓	Multi-view anchor-graph clustering
Proposed CCM	✓	✓	✓	Dynamic	✓	✓	YOLO + LSTM + BERT + 3DCNN

Building on these advancements, this paper proposes a novel approach that combines object detection, skeleton-based, and subtitle-based processing techniques to enhance the accuracy of action recognition in video identification.

Table 1 compares the related studies and proposed CCM. The 'RGB,' 'Skeleton,' and 'Subtitles' denote analyzed features. The 'Pattern' differentiates static and dynamic approaches. The 'Temporal' and 'Spatial' represent whether the mechanism considers the temporal and spatial features, respectively. The 'Models' specifies the used models. In contrast to related studies, the proposed CCM employs an array of powerful models, including YOLO, LSTM, BERT, and 3DCNN models, for extracting and analyzing diverse features from various possible perspectives.

3 Notations, assumptions, and problem descriptions

This section introduces notations, assumptions, problem descriptions, and objectives of this paper.

3.1 Notations and assumptions

Assume that there is a baseball video, denoted by $V = (f_1, f_2, ..., f_m)$, which consists of m continuous frames, where f_i represents $i\text{-}th$ frame of V. Each frame f_i contains the played time point, subtitles, and objects, which are denoted by (t_i, s_i, o_i). Notation t_i denotes the time point at which the frame f_i appears while notation s_i represents the subtitles contained within the frame f_i which may convey important information to the viewers. The notation o_i denotes the objects, which make up the visual elements of the scene. The objects may consist of a pitcher, a batter, a ball, a bat, some catchers, umpires, gloves, helmets, or other elements presented in the baseball video.

Let $H = (V_1, V_2, ..., V_h)$ denote the set of h strikeout highlights in V. Let $[t_{i,j}, t_{i,k}]$ and $[\widehat{t_{i,j}}, \widehat{t_{i,k}}]$ denote the labeled and predicted start and end time points of the i-th strikeout highlight V_i, respectively. Let I_i and U_i denote the intersection and union of $[t_{i,j}, t_{i,k}]$ and $[\widehat{t_{i,j}}, \widehat{t_{i,k}}]$, respectively. The values of I_i and U_i can be calculated by Eqs. (1) and (2), respectively.

$$I_i = \left[t_{i,j}, t_{i,k}\right] \cap \left[\hat{t}_{i,j}, \hat{t}_{i,k}\right] \tag{1}$$

$$U_i = [t_{i,j}, t_{i,k}] \cup [\hat{t}_{i,j}, \hat{t}_{i,k}] \tag{2}$$

Let ρ_i denote the Intersection Over the Union (IoU) of the i-th strikeout highlight. Notation ρ_i aims to measure the overlap between the predicted and labeled time intervals. The value of ρ_i can be calculated by Eq. (3).

$$\rho_i = \frac{I_i}{U_i} \tag{3}$$

3.2 Problem descriptions

Let ρ_{th}^{TP} denote the true positive threshold of IoU. Let TP_i^M denote true positive of the time interval of the i-th strikeout highlight predicted by mechanism M. That is, the intersection of $[t_{i,j}, t_{i,k}]$ and $[\widehat{t_{i,j}}, \widehat{t_{i,k}}]$ for the strikeout behavior is

correct. The value of TP_i^M can be calculated by Eq. (4).

$$TP_i^M = \begin{cases} 1, \ \rho_i \geq \rho_{th}^{TP}, \\ 0, \ otherwise. \end{cases} \tag{4}$$

Similarly, let FP_i^M and FN_i^M denote false positive and false negative of the predicted time interval $[\widehat{t_{i,j}}, \widehat{t_{i,k}}]$, respectively. Let ρ_{th}^{FP}, ρ_{th}^{FN} denote the false positive and false negative threshold of IoU, respectively. The values of FP_i^M and FN_i^M can be calculated by Eqs. (5) and (6), respectively.

$$FP_i^M = \begin{cases} 1, \ \frac{U_i \setminus [t_{i,j}, t_{i,k}]}{U_i} \geq \rho_{th}^{FP}, \\ 0, \quad otherwise, \end{cases} \tag{5}$$

$$FN_i^M = \begin{cases} 1, \ \frac{U_i \setminus [\widehat{t_{i,j}}, \widehat{t_{i,k}}]}{U_i} \geq \rho_{th}^{FN}, \\ 0, \quad otherwise. \end{cases} \tag{6}$$

Let TP^M, FP^M and FN^M denote true positive, false positive, and false negative of h strikeout highlights in V by applying mechanism M, respectively. The values of TP^M, FP^M and FN^M can be calculated by Eqs. (7) ~ (9), respectively.

$$TP^M = \sum_{i=1}^{h} TP_i^M \tag{7}$$

$$FP^M = \sum_{i=1}^{h} FP_i^M \tag{8}$$

$$FN^M = \sum_{i=1}^{h} FN_i^M \tag{9}$$

Let P^M and R^M denote the precision and recall of the existing strikeout highlights in the video V using mechanism M, respectively. The values of P^M and R^M can be calculated by Eqs. (10) and (11), respectively.

$$P^M = \frac{TP^M}{TP^M + FP^M} = \sum_{i=1}^{h} TP_i^M / (\sum_{i=1}^{h} TP_i^M + \sum_{i=1}^{h} FP_i^M) \tag{10}$$

$$R^M = \frac{TP^M}{TP^M + FN^M} = \frac{\sum_{i=1}^{h} TP_i^M}{\sum_{i=1}^{h} TP_i^M + \sum_{i=1}^{h} FN_i^M} \tag{11}$$

Let $F1^M$ denote the $F1$-score of the existing strikeout highlights in the video V using mechanism M. The value of $F1^M$ can be calculated by Eq. (12).

$$F1^M = \frac{2 * P^M * R^M}{P^M + R^M} \tag{12}$$

3.3 Objectives

Let Φ present the set of all possible mechanisms that are used to identify strikeout highlights. The primary objective of this paper is to discover the optimal mechanism $\mathcal{M}_1$ that fulfills the requirement specified in Eq. (13).

3.3.1 Objective function 1:

$$\mathcal{M}_1 = \underset{M \in \Phi}{argmax} F1^M \tag{13}$$

If multiple mechanisms satisfy the criteria of objective function 1, the second objective of this paper is to find the optimal mechanism $\mathcal{M}_2$ that exhibits the highest IoU between the labeled and predicted time intervals. Let ρ_i^M denote the IoU of i-th strikeout highlight applying mechanism M. The second objective function is presented in Eq. (14).

3.3.2 Objective function 2:

$$\mathcal{M}_2 = \underset{M \in \mathcal{M}_1}{arg\,max} \sum_{i=1}^{h} \rho_i^M \tag{14}$$

When achieving the abovementioned objectives, it is essential to meet certain constraints. Let T denote the total period of video V. Let the Boolean variable σ_t denote whether the frame f_t is a strikeout highlight. That is,

$$\sigma_t = \begin{cases} 1, \text{ if } f_t \text{is strikeout highlight}, \\ 0, \qquad \text{otherwise}. \end{cases}$$

This constraint, as expressed in Eq. (15), ensures that the strikeout highlight is a valid subset of frames or spatial elements existing in the video.

1) Spatial constraint

$$\exists T_i = \left[t_{i,j}, \ t_{i,k}\right] \subseteq T \text{ Such that}$$

$$\prod_{t=j}^{k} \sigma_t = 1 \tag{15}$$

In baseball, a strikeout, often referred to as a 'K' in scorekeeping, occurs when a batter fails to contact the ball after receiving three strikes. A strike is recorded when a pitched ball passes through the strike zone denoted by $\mathcal{Z}$ and the batter either does not attempt to swing denoted by $\mathcal{N}$, or swings and misses the ball denoted by $\mathcal{S}$. Let $r_{i,j}$ denote whether

the outcome of the i-th pitching is a strike for some specific batter b_j. That is,

$$r_{i,j} = \begin{cases} 1, \mathcal{Z}\&(\mathcal{N}|\mathcal{S})istrue, \\ 0, \quad \text{otherwise.} \end{cases}$$

The following defines the necessary features of strikeout which have been generally known in the baseball rules.

2) Features constraint

Let $V_k \in V$ be a segment of a baseball video. The V_k is said to be a strikeout highlight if it has the following well-known features at least.

(a) The pitcher delivers a ball.
(b) At least three consecutive pitches are satisfying the condition:

$$r_{i+1,j} * r_{i+2,j} * r_{i+3,j} = 1$$

The following constraint, as shown in Eq. (16), guarantees that there are different features between the strikeout highlights and non-strikeout videos in video$V$. Let $T_i = [t_{i,a}, t_{i,b}]$ denote the labeled strikeout highlight of V_i. Let $T_j = [t_{j,a}, t_{j,b}]$ denote the labeled non-strikeout video of V_j, where $T_i \cap T_j = \varnothing$. Additionally, let $F(V)$ denote any feature extracting function capable of extracting features from a video V.

3) Feature-difference constraint

$$\exists V_i^{sub} \in V_i and V_j^{sub} \in V_j such that$$

$$F\left(V_i^{sub}\right) \neq F\left(V_j^{sub}\right) \tag{16}$$

4 The proposed CCM mechanism

This paper proposes a strikeout identification mechanism, called CCM, which aims to identify highlights of strikeout moments in baseball videos. The architecture of CCM is depicted in Fig. 1. The proposed CCM consists of two phases: *coarse-grain clipping for candidate videos* and *fine-grain identification for strikeout highlights*. In the first phase, the proposed CCM adopts YOLO and LSTM models to coarsely clip the candidate videos. The second phase employs YOLO, 3DCNN, and BERT models to identify the strikeout highlights. The subsequent sections provide further details on these two phases.

4.1 Coarse-grain clipping for candidate videos

The primary objective of this phase is to identify a video segment in a baseball video where the batter is recorded getting out on the final pitch. This segment, spanning from the pitcher's pitching to the batter's out, is referred to as the 'candidate' video. Although it represents the last pitch resulting in the batter's out, it is important to emphasize that the out could be a result of a caught ball rather than a strikeout. Therefore, in the subsequent fine-grained processing phase, further meticulous examination of this segment is required to determine if it indeed represents the desired strikeout segment.

The proposed CCM mechanism employs YOLOv5s and LSTM models to identify whether a given video contains strikeout highlights. Initially, the proposed mechanism uses YOLOv5s to observe the various out-indicators to determine the end time of candidate videos. Additionally, it employs YOLOv5s to detect the 'person' objects and utilizes LSTM to analyze the skeleton information of each 'person' object for determining the start time of the candidate videos.

Initially, baseball videos have been collected. These videos are annotated using LabelImg as the training data for supervised learning. The original YOLOv5s was implemented on the COCO2017 dataset that had 80 categories [14]. To reduce the number of training parameters, the annotation is executed based on the seven categories, aiming to ensure high detection accuracy. The annotation category information of YOLO is shown in Table 2.

Figure 2 illustrates the architecture of YOLOv5s, which is composed of the backbone, neck, and prediction. The backbone is responsible for extracting relevant feature maps from the input frames, enhancing object detection effectively. It comprises several essential components, including Focus, CBL (Convolutional Block Layer), CSP (Cross Stage Partial), and SPP (Spatial Pyramid Pooling). These components play crucial roles in extracting informative features from input frames, enabling the YOLOv5s model to accurately detect objects. The neck is responsible for gathering the extracted features from the backbone and concatenating them to generate a feature pyramid, aiming to recognize objects of various sizes and scales within the same category. The neck utilizes the convolutional layers to generate the final predictions. The predictions output three pyramid levels, which are small (152*152*255), medium (304*304*255), and large scales (608*608*255).

The YOLOv5s model is used to predict bounding boxes for the predefined categories. For each input frame, the model generates a vector associated with each predicted bounding box. Let $\mathbb{V}_{i,j}^{Y}$ denote the YOLOv5s vector for the j-th predicted bounding box of the frame f_i. The components of $\mathbb{V}_{i,j}^{Y}$ can be represented by Eq. (17).

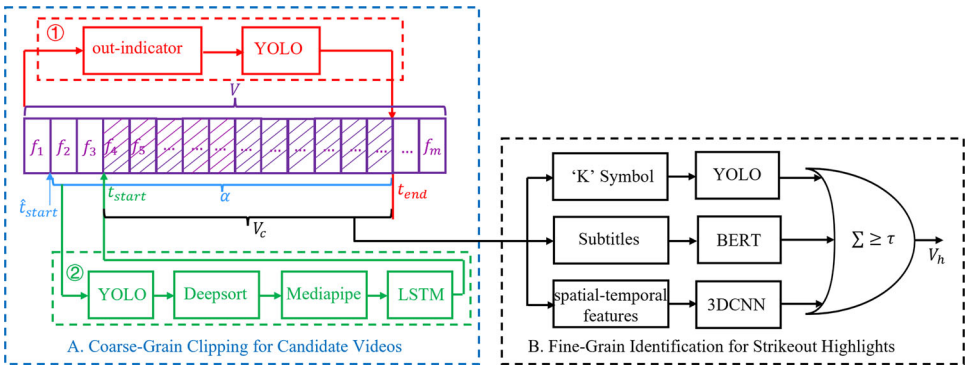


Fig. 1 The architecture of the proposed CCM

Table 2 Annotation category information of YOLO

Name	Category number
person	0
ball	1
bat	2
glove	3
helmet	4
out-indicator	5
K	6

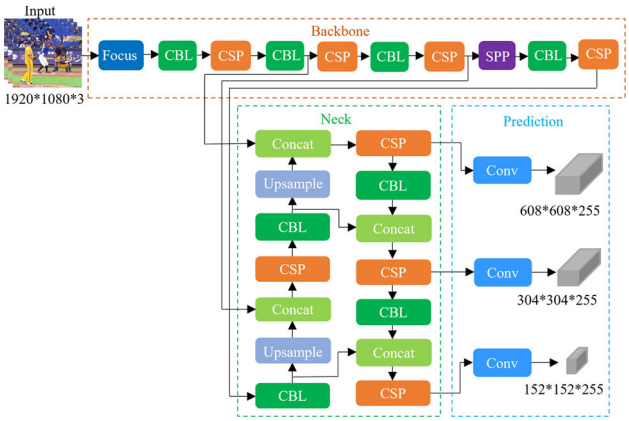


Fig. 2 The architecture of YOLOv5s

$$\mathbb{V}_{i,j}^{Y} = \left(p_{i,j},\, x_{i,j},\, y_{i,j},\, w_{i,j},\, h_{i,j},\, c_{i,j}^{0},\, c_{i,j}^{1},\, ...,\, c_{i,j}^{6}\right) \tag{17}$$

where $p_{i,j}$ denotes the probability that the object exists in the j-th predicted bounding box of the frame f_i, which can be either 1 or 0. Notations $x_{i,j}$ and $y_{i,j}$ denote the center point coordinates of the j-th predicted bounding box, while $w_{i,j}$ and $h_{i,j}$ denote its width and height, respectively. The notations $c_{i,j}^{0}$, $c_{i,j}^{1}$, ..., $c_{i,j}^{6}$ denote the probabilities of the categories outlined in Table 2.

The loss function uses CIoU (Complete-IoU), which considers the IoU, scale, penalty term, and distance between the labeled and predicted bounding boxes. Let $\mathbb{I}_{i,j}$ and $\mathbb{U}_{i,j}$ denote the intersection and union between the j-th labeled and predicted bounding boxes of the frame f_i, respectively. Let $\varphi_{i,j}$ and $\psi_{i,j}$ denote the IoU and CIoU of the j-th predicted bounding box for the frame f_i, respectively. The values of $\varphi_{i,j}$ and $\psi_{i,j}$ can be calculated by Eqs. (18) and (19), respectively.

$$\varphi_{i,j} = \frac{\mathbb{I}_{i,j}}{\mathbb{U}_{i,j}} \tag{18}$$

$$\psi_{i,j} = \varphi_{i,j} - \frac{\mathbb{c}_{i,j}^{2}}{\mathbb{d}_{i,j}^{2}} - \frac{\upsilon_{i,j}^{2}}{1 - \varphi_{i,j} + \upsilon_{i,j}} \tag{19}$$

where $\mathbb{c}_{i,j}$ denotes the Euclidean distance between the center points of the j-th labeled and predicted bounding boxes. The $\mathbb{d}_{i,j}$ denotes the diagonal distance of the minimum bounding rectangle that can contain both the j-th labeled and predicted bounding boxes. The $\upsilon_{i,j}$ denotes the consistency of the aspect ratio between the j-th labeled and predicted bounding boxes. The value of $\upsilon_{i,j}$ can be calculated by Eq. (20).

$$\upsilon_{i,j} = \frac{4}{\pi^2}\left(arctan\frac{w_{i,j}}{h_{i,j}} - arctan\frac{\hat{w}_{i,j}}{\hat{h}_{i,j}}\right)^2 \tag{20}$$

where $w_{i,j}$ and $h_{i,j}$ denote the width and height of the j-th labeled bounding box, respectively. Similarly, $\widehat{w}_{i,j}$ and $\widehat{h}_{i,j}$ denote the width and height of the j-th predicted bounding box, respectively.

Let $\mathcal{L}_{i,j}^{cIoU}$ denote the loss function for the j-th labeled and predicted bounding boxes. The value of $\mathcal{L}_{i,j}^{cIoU}$ can be calculated by Eq. (21).

$$\mathcal{L}_{i,j}^{cIoU} = 1 - \psi_{i,j} \tag{21}$$

4.1.1 Identifying the end time of candidate videos

Since the end time of the candidate video has the feature that the out-indicator will be changed, the YOLOv5s model is utilized to identify the number shown in the out-indicator. The end time of a candidate video mainly observes the change of out-indicator, specifically the transition from 2-out to 0-out, which corresponds to the 5-th category of the YOLOs model, as shown in Table 1. Let t_{end} denote the end time point of a candidate video. Let c_{th}^{k} denote the probability threshold of the k-th category, where $0 \leq k \leq 6$. The t_{end} is corresponding to the i-th frame with the minimum loss in the j-th bounding box where the probability $c_{i,j}^{5}$ is greater than the c_{th}^{5}. It can be derived from Eq. (22).

$$t_{end} = arg \min_{\forall i, c_{i,j}^{5} \geq c_{th}^{5}} \mathcal{L}_{i,j}^{cIoU} \tag{22}$$

To extend the candidate video, a forward extension is necessary to ensure it spans from the pitcher's pitch until the end of the inning. This entails searching α time slots back from the end timestamp t_{end} to locate a video segment of a specific duration for candidate video consideration. The value of α is determined based on the editorial experience and observations from the dataset. This duration considers the period from the pitcher's preparation to the batter's out. Let $\widehat{t}_{start}$ denote the extended time point of a candidate video. The value of $\widehat{t}_{start}$ can be denoted by Eq. (23).

$$\widehat{t}_{start} = t_{end} - \alpha \tag{23}$$

Let V_{temp} denote the video with the time intervals $[\widehat{t}_{start}, t_{end}]$. Since V_{temp} may not start from the pitcher's pitching, the YOLOv5 and LSTM models are employed to further identify the start time of the candidate videos.

4.1.2 Determining the start time of candidate videos

Given the candidate video that starts with the pitcher's pitching, this step utilizes YOLOv5 and LSTM to identify the frame that contains the pitcher. However, due to the potential similarity in clothing among the pitcher and their teammates, YOLO faces difficulties in exclusively detecting the pitcher. Therefore, the proposed mechanism uses YOLOv5s to detect 'person' objects in general, rather than specifically focusing on identifying the pitcher. There may be multiple 'person' objects in a frame. Then the proposed mechanism further adopts Deep Simple Online Realtime Tracking (DeepSORT) to assign a unique ID for each 'person' object and individually track them. The IDs can be used to distinct different 'person' objects in a frame. Subsequently, individuals with the same ID are transformed into skeletons using Mediapipe [36]. The proposed CCM employs the LSTM model which has been well-trained to identify the pitcher's pitching based on the time-sequence skeletons of each 'person' object. By analyzing the skeletal data, the LSTM model identifies the pitcher while the start time of the candidate video can be obtained.

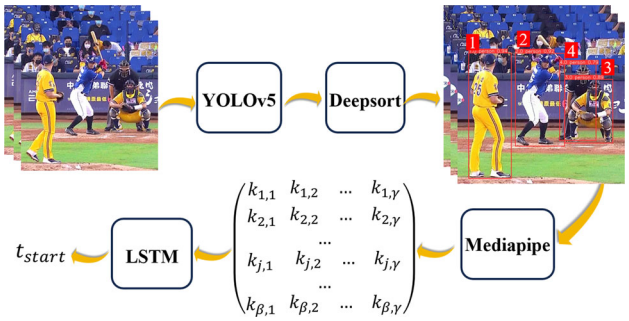


Fig. 3 The process of determining the start time of candidate videos

Figure 3 illustrates an example to show how the proposed CCM determines the start time of the candidate video. Initially, the YOLOv5s is utilized to detect the 'person' objects in the frame f_i. Subsequently, the proposed CCM employs DeepSORT assigning unique IDs to the detected bounding boxes of the 'person' objects. Then each 'person' object with the same ID is transformed into skeletons by Google Mediapipe. Assume that there are γ frames in the video V_{temp}, generated in the previous step. Let β denote the number of IDs. Let $K_j = (k_{j,1}, ..., k_{j,i}, ..., k_{j,\gamma})$ denote a continuous set of skeleton coordinates of the person with ID=j in the video V_{temp}. The $k_{j,i}$ serves as input to LSTM. Let $\widehat{y}_{j,i}$ and y_i denote the predicted and labeled pitcher's pitching behaviors, respectively. The loss function L^{LSTM} of the LSTM model can be defined as Exp. (24).

$$L^{LSTM}(y_i, \hat{y}_{j,i}) = -[y_i * \log(\hat{y}_{j,i}) + (1 - y_i) * \log(1 - \hat{y}_{j,i})] \tag{24}$$

Let t_{start} denote the start time of a candidate video. The t_{start} corresponds to the i-th frame with minimal loss function among all 'person' objects. It can be derived by Eq. (25).

$$t_{start} = \arg \min_{1 \leq i \leq \gamma, 1 \leq j \leq \beta} L^{LSTM}(y_i, \hat{y}_{j,i}) \tag{25}$$

Let V_c and T_c denote the candidate video and its time interval, respectively. The values of V_c and T_c can be derived from Eq. (26) and (27), respectively.

$$V_c = [f_{start}, f_{start+1}, ..., f_{end}] and \tag{26}$$

$$T_c = [t_{start}, t_{end}] \tag{27}$$

where f_{start} and f_{end} denote the frames corresponding to t_{start} and t_{end}, respectively.

Till now, a candidate video V_c has been obtained, starting with the pitcher's pitch and ending with the out-indicator changing from 2 to 0. The next phase needs to further identify whether the candidate video is the expected strikeout highlight.

4.2 Fine-grain identification for strikeout highlights

Although the candidate video captures the last pitch before the batter's out, it does not necessarily guarantee a strikeout. To address this, this phase employs YOLOv5, BERT, and 3DCNN models to identify whether the candidate video is a strikeout highlight. Herein, it is emphasized that each of the three models can recognize whether the candidate video is a strikeout highlight. However, they each have their strengths and weaknesses. To optimize recognition accuracy, the proposed CCM integrates and utilizes these three models together, complementing each other's advantages and disadvantages.

4.2.1 Identifying strikeout highlights using YOLOv5

This step employs YOLOv5s to identify the presence of the letter 'K' in candidate videos, which corresponds to the 6-th category as shown in Table 1. The YOLOv5s is used to identify the j-th bounding box in the frame $f_i \in V_c$ and determine whether the output $c^6_{i,j}$ is greater than c^6_{th}. Let ξ_{YOLO} be a Boolean variable indicating whether V_c is a strikeout highlight based on YOLOv5s. It can be derived from Eq. (28).

$$\xi_{YOLO} = \begin{cases} 1, \exists f_i \in V_c \text{ such that } c^6_{i,j} \geq c^6_{th}, \\ 0, \qquad\qquad \text{otherwise.} \end{cases} \tag{28}$$

4.2.2 Identifying strikeout highlights using BERT

This step employs the pre-trained language model BERT to analyze the subtitles in the candidate video V_c, aiming to check if it is related to 'strikeout'. Recall that the subtitles in the frame f_i are represented by s_i. Let $S_c = (s_1, s_2, ..., s_k)$ denote a continuous set of k subtitles of the candidate's video V_c. Each s_i is divided into a set of words applying the tool CKIP (Chinese Knowledge and Information Processing Toolkit). That is.

$$s_i = (w_{i,1}, w_{i,2}, ..., w_{i,|s_i|})$$

As shown in Fig. 4, each $s_i \in S_c$ prepended with a start token [CLS] (Classification), serves as an input to the BERT model. The input length of BERT corresponds to the maximal word length of s_i. A Binary classifier in the downstream network employs the input subtitles to determine whether the output is the 'strikeout'. Let $\widehat{b_i}$ and b_i denote the predicted

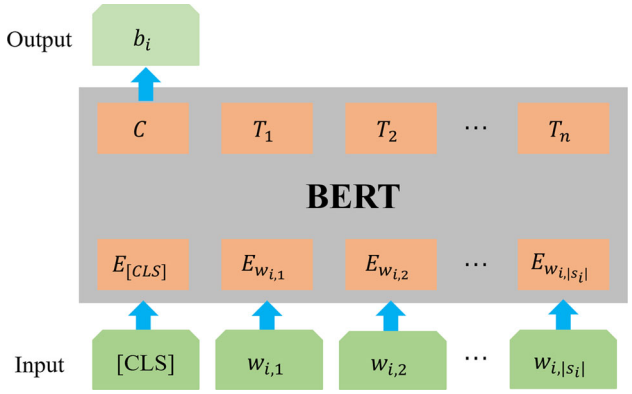


Fig. 4 The procedure of BERT

and labeled 'strikeout', respectively. Let L^{BERT} denote the loss function of the BERT model. The value of L^{BERT} can be derived from Eq. (29).

$$L^{BERT}(b_i, \widehat{b_i}) = -[b_i * \log(\widehat{b_i}) + (1 - b_i) * \log(1 - \widehat{b_i})] \tag{29}$$

Let η^{BERT} denote the threshold value at which the predicted $\widehat{b_i}$ is considered as a strikeout. Let ξ_{BERT} be a Boolean variable which indicates whether V_c is a strikeout highlight determined by the BERT model. It can be derived from Eq. (30).

$$\xi_{BERT} = \begin{cases} 1, \exists s_i \in S_c \text{ such that } L^{BERT}(b_i, \widehat{b}) \geq \eta^{BERT}, \\ 0, \qquad\qquad\qquad \text{otherwise.} \end{cases} \tag{30}$$

Till now, the proposed CCM has applied the BERT model to check the subtitles and determine whether the candidate video is a strikeout highlight. The next subsection will propose the 3DCNN model to further check if V_c is a strikeout highlight.

4.2.3 Identifying strikeout highlights using 3DCNN

This step adopts 3DCNN to identify the candidate video V_c by learning short-term spatial–temporal features associated with 'strikeout'. Recall that a constraint for a strikeout is to satisfy the condition $\mathcal{Z}\&(\mathcal{N}|\mathcal{S})$. Herein, the 3DCNN is applied as a binary classifier where the former layers extract the features of V_c and the later layers classify whether the features of V_c match the 'strikeout' behavior. The proposed CCM considers a set of ð frames per second in the candidate video V_c. Each frame $f_i \in V_c$ with dimensions of 32*32, serves as input to the 3DCNN model. The architecture of the 3DCNN model consists of Conv3D_1 to Conv3D_3 layers, with specific kernel size, stride, padding, and output size outlined in Table 3.

Table 3 The architecture of the 3DCNN model

Layers	Blocks	Output size
Input	32*32*15*3	
Conv3D_1	kernel_size (3, 3, 3) stride (1, 1, 1) padding (1, 1, 1)	32*32*15*32
Pooling3D_1	kernel_size (12, 12, 7) stride (2, 2, 2)	11*11*5*32
Conv3D_2	kernel_size (3, 3, 3) stride (1, 1, 1) padding (1, 1, 1)	11*11*5*64
Conv3D_3	kernel_size (3, 3, 3) stride (1, 1, 1) padding (1, 1, 1)	11*11*5*64
Pooling3D_2	kernel_size (5, 5, 3) stride (2, 2, 2)	4*4*2*64
Flatten	–	2048
Dense_1	–	512
Dropout	0.75	128
Dense_2	–	128

Let $\widehat{d}$ and d denote the predicted and labeled 'strikeout', respectively. Let L^{3DCNN} denote the loss function of the 3DCNN model. The value of L^{3DCNN} can be derived from Eq. (31).

$$L^{3DCNN}\left(d, \widehat{d}\right) = -\left[d * \log(\widehat{d}) + (1-d)\left(\log(1-\widehat{d})\right)\right] \tag{31}$$

Let η^{3DCNN} denote the threshold value at which the candidate V_c is considered as a strikeout. Let ξ_{3DCNN} be a Boolean variable which indicates whether V_c is a strikeout highlight based on 3DCNN. It can be derived from Eq. (32).

$$\xi_{3DCNN} = \begin{cases} 1, L^{3DCNN}\left(d, \widehat{d}\right) \geq \eta^{3DCNN}, \\ 0, \qquad \text{otherwise.} \end{cases} \tag{32}$$

Till now, this paper has proposed three models, including YOLOv5, BERT, and 3DCNN, which aim to check if the candidate video V_c is a strikeout highlight. The next step aims to integrate the three approaches and make a final decision.

The integration of three approaches checks whether the candidate's video V_c is a strikeout highlight V_h, as shown in Fig. 5. If the sum of the results from the three models is greater than or equal to the threshold value τ, the candidate video V_c is categorized as a strikeout highlight V_h. Let $\xi_{highlight}$ be a Boolean variable indicating whether V_c is a strikeout highlight. The value of $\xi_{highlight}$ can be derived from Eq. (33).

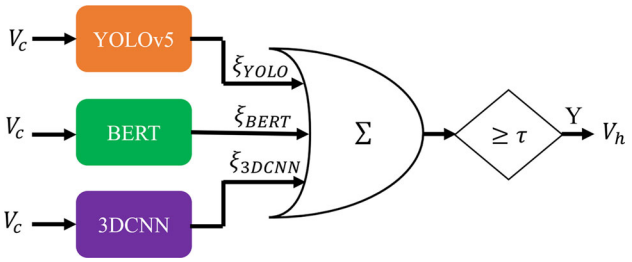


Fig. 5 The integration of three models, including YOLOv5, BERT, and 3DCNN

$$\xi_{highlight} = \begin{cases} 1, if\left(\xi_{YOLO} + \xi_{BERT} + \xi_{3DCNN}\right) \geq \tau, \\ 0, \qquad \text{otherwise.} \end{cases} \tag{33}$$

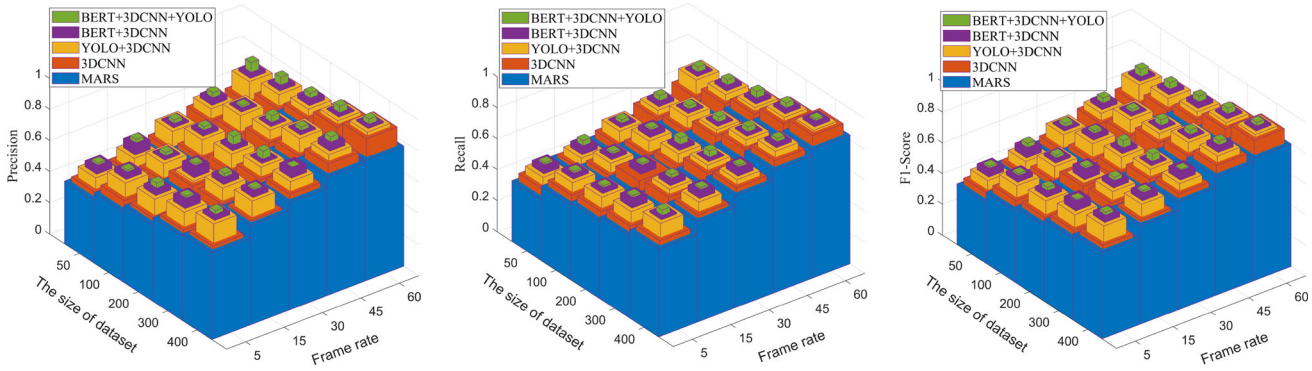
5 Performance evaluation

This section aims to assess the performance enhancement of the proposed CCM in comparison to the MARS (Motion-Augmented RGB Stream) [37]. The MARS was developed based on the principles of distillation and learning with privileged information, aiming to avoid flow computation at test time, while maintaining the performance of two-stream approaches. However, its performance might be limited when dealing with highly complex or diverse behavior recognition tasks. This study employs various models to implement fine-grained strategies, including 3DCNN, BERT + 3DCNN, YOLO + 3DCNN, and BERT + 3DCNN + YOLO models. The following presents the datasets and simulation results.

5.1 Datasets

The evaluation is conducted based on the ELTA dataset, a custom dataset collected in collaboration with ELTA, with video clips sourced from the ELTA platform. Figure 6 showcases sample videos from this dataset. The dataset comprises 800 video clips encompassing various baseball activities, including 413 strikeout highlight videos and 387 non-strikeout videos. It is divided into a training set, which contains 328 strikeout videos and 312 non-strikeout videos, and a test set, consisting of 85 strikeout videos and 75 non-strikeout videos. It is worth noting that the ELTA dataset provides frame-level annotations for each video. This detailed annotation approach enables the model to perform precise analysis and identification on every individual frame.

Fig. 6 Sample videos from the ELTA dataset



(a) Precision (b) Recall (c) F1-Score

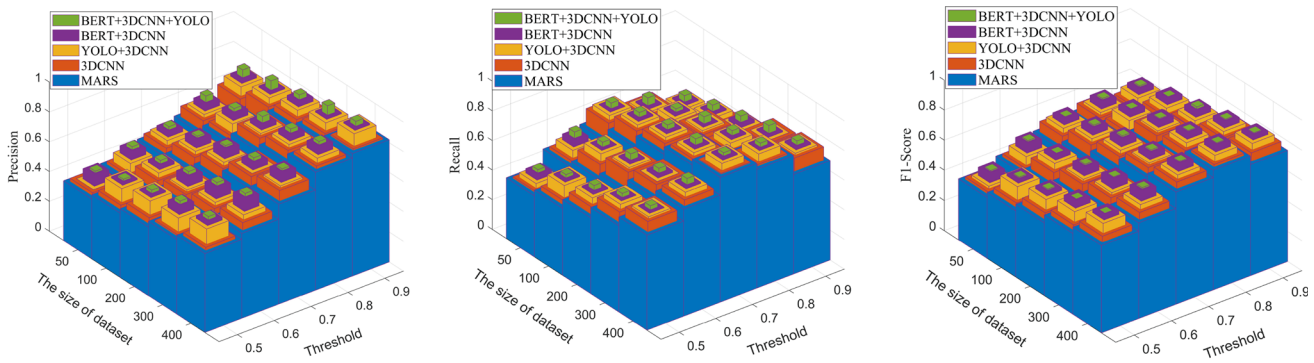
Fig. 7 Comparison of various models by varying the frame rate and dataset size

5.2 Simulation results

Figure 7a, b and c compare the MARS, 3DCNN, BERT + 3DCNN, YOLO + 3DCNN and BERT + 3DCNN + YOLO in terms of precision, recall, and F1-Score. This comparison is performed by varying dataset size and frame rate. The dataset size ranges from 50 to 400, and the frame rate varies from 5 to 60 per second. All the compared models have the common trend that the precision, recall, and F1-Score are increasing with the frame rate. The reason is that a higher frame rate provides more temporal information, enabling models to capture fast-paced behaviors and subtle movements more accurately, ultimately improving their performance. Furthermore, the precision, recall, and F1-score are increased with the size of the dataset. This is because that a larger amount of training data helps the models improve the learning capacities and effectively capture the inherent features within the dataset.

In comparison, the BERT + 3DCNN + YOLO model exhibits superior precision, recall, and F1-Score compared to the other compared models due to its capability to leverage the strengths of YOLO, 3DCNN, and BERT, effectively identifying strikeout highlights by combining heterogeneous features. This combination enables the model to make more precise predictions. A segment of the video is to be identified as the strikeout only when the three models report 'Positive'. These strict conditions increase the opportunities for false negatives but might miss the opportunities for true positives. As a result, it achieves lower recall and F1-Score.

Figure 8a, b, and c present a comparative analysis of MARS, 3DCNN, BERT + 3DCNN, YOLO + 3DCNN, and BERT + 3DCNN + YOLO in terms of precision, recall, and



(a) Precision (b) Recall (c) F1-Score

Fig. 8 Comparison of various models by varying the threshold and dataset size

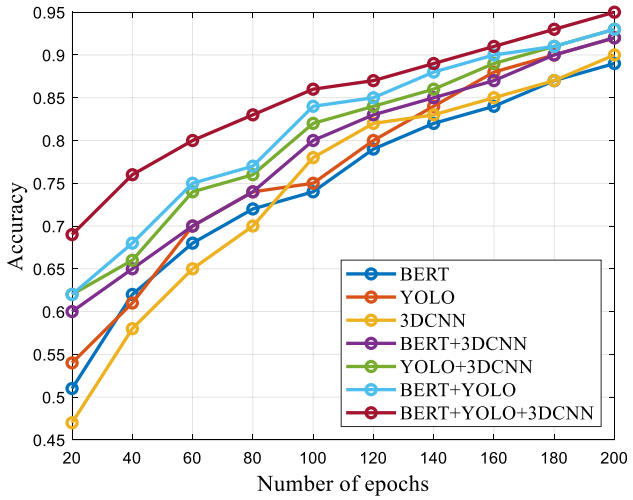


Fig. 9 Comparison of various models in terms of accuracy over different numbers of epochs

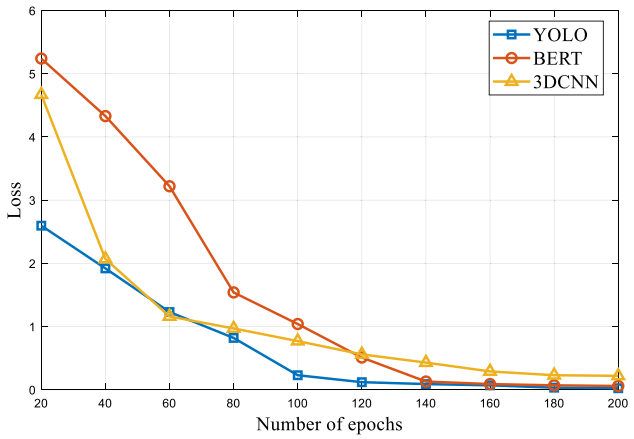


Fig. 10 Comparison of various models in terms of loss over varying epochs

F1-Score by varying dataset size and threshold. The dataset size ranges from 50 to 400, and the threshold ranges from 0.5 to 0.9, serving to classify strike highlights or non-highlights. All the compared models have the common trend that the precision is increased with the threshold. However, the recall and F1-score increase with the threshold when the threshold value is below 0.7 but decrease when the threshold exceeds 0.8. This is because when the threshold is larger than 0.8, an output with a value of 0.75 is a highlight but the system considers it is non-highlight, which results in a false negative. Consequently, the value of recall and F1-score are reduced. Figure 9 compares the 3DCNN, YOLO, BERT, and their combinations in terms of accuracy under different numbers of epochs. The number of epochs is ranging from 20 to 200. A common trend is that the accuracy increases with the epochs. The reason is that an increased number of training epochs can enhance the model's capability to learn the features of the data, resulting in improved predictive ability. The proposed BERT + YOLO + 3DCNN outperforms the other models. The reason is that the model employs a powerful and holistic approach to capturing diverse features by integrating the strengths of BERT, YOLO, and 3DCNN, ultimately leading to enhanced accuracy in behavior recognition tasks.

Figure 10 compares the loss values of the YOLO, BERT, and 3DCNN models in terms of the number of epochs using the Exps. (24), (29), and (31), respectively. The number of epochs is ranging from 20 to 200. In the early epochs, the model may experience significant fluctuations in loss as it starts to learn the underlying patterns within the data. However, as the training continues, the model converges towards an optimal set of parameters, leading to a smoother and more gradual reduction in loss.

Figure 11 illustrates the impact of the number of epochs and the learning rate on the accuracy of the proposed CCM during the training process. The number of epochs varies ranging from 50 to 200, and the learning rate varies ranging from 10^{-4} to 1. The accuracy is increased with the number of epochs. The reason is that with more training epochs,

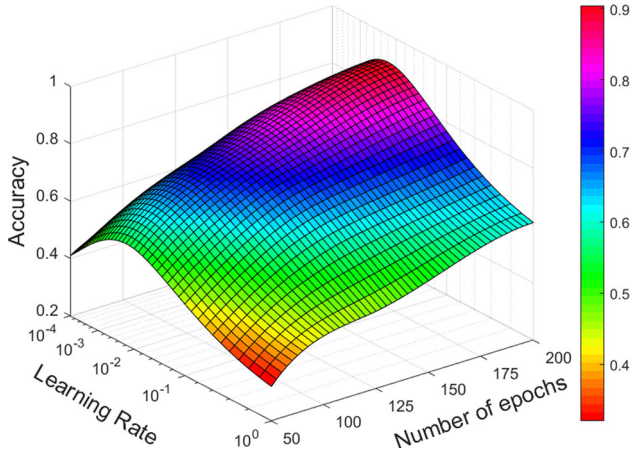


Fig. 11 The accuracy by varying epochs and learning rate

the model can better learn the features of the data, improving its predictive ability. On the other hand, the accuracy is also increased with the learning rate, while the accuracy is decreasing with the learning rate exceeding 0.1. The reason is that excessively large learning rates may lead to unstable training processes, causing oscillations or divergence.

6 Conclusions

This paper introduces a novel behavior identification mechanism, called CCM, designed to proficiently detect 'strikeout' highlights in baseball video. The proposed CCM is a fusion of diverse deep learning techniques, including YOLO for object detection, BERT for subtitle analysis, LSTM for processing skeletal features, and 3DCNN for identifying 'strikeout' highlights. The simulation results exhibit the remarkable performance of our proposed CCM when compared to existing models, demonstrating its efficacy in behavior recognition tasks. Future work will focus on further refining and fine-tuning the CCM mechanism to achieve optimal performance in behavior recognition tasks.

Acknowledgements This work was supported by the Higher Education Research Program Project under Grant No. 2022AH010067, the Anhui Provincial Natural Science Foundation under Grant No. 2408085MF177, the Research Projects of the Chuzhou University under Grant No. 2022XJZD14, the Anhui Province's University Research Project under Grant No. 2023AH040216, and the Anhui Outstanding Youth Fund Project under Grant No. 2022AH030109.

Author contributions Conceptualization: Qiaoyun Zhang, Chih-Yung Chang; Methodology: Qiaoyun Zhang, Chih-Yung Chang, Hsiang-Chuan Chang, Cuijuan Shang; Formal analysis: Qiaoyun Zhang, Chih-Yung Chang; Writing—original draft preparation: Qiaoyun Zhang, Chih-Yung Chang; Writing—review and editing: Cuijuan Shang, Diptendu Sinha Roy.

Data availability The datasets generated during the current study are not publicly available but are available from the corresponding author.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Li, D., Yao, T., et al.: Unified spatio-temporal attention networks for action recognition in videos. IEEE Trans. Multimed. **21**(2), 416–428 (2019)
2. Yang, Z., Li, Y., et al.: Action recognition with spatio–temporal visual attention on skeleton image sequences. IEEE Trans. Circuits Syst. Video Technol. **29**(8), 2405–2415 (2019)
3. Monfort, M., Pan, B., et al.: Multi-moments in time: learning and interpreting models for multi-action video understanding. IEEE Trans. Pattern Anal. Mach. Intell. **44**(12), 9434–9445 (2022)
4. Sharma, V., Gupta, M., et al.: A review of deep learning-based human activity recognition on benchmark video datasets. Appl. Artif. Intell. **36**(1), 2855–2901 (2022)
5. R. Girshick, J. Donahue, et al, 2014. "Rich feature hierarchies for accurate object detection and semantic segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, USA, pp. 580–587
6. R. Girshick, 2015. "Fast r-cnn," *IEEE International Conference on Computer Vision*, Beijing, China, pp. 1440–1448
7. Zhao, H., Li, Z., et al.: Attention based single shot multibox detector. J. Electron. Inf. Technol. **43**(7), 2096–2104 (2021)
8. J. Redmon, S. Divvala, et al, 2016. "You only look once: unified, real-time object detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, Washington, USA, pp. 779–788
9. Liang, S., Wu, H., et al.: Edge YOLO: real-time intelligent object detection system based on edge-cloud cooperation in autonomous vehicles. IEEE Trans. Intell. Transp. Syst. **23**(12), 25345–25360 (2022)
10. Song, Y., Xie, Z., et al.: MS-YOLO: object detection based on YOLOv5 optimized fusion millimeter-wave radar and machine vision. IEEE Sens. J. **22**(15), 15435–15447 (2022)
11. Shi, Y., Tian, Y., et al.: Sequential deep trajectory descriptor for action recognition with three-stream CNN. IEEE Trans. Multimedia **19**(7), 1510–1520 (2017)
12. Liu, J., Shahroudy, A., et al.: NTU RGB+D: a large-scale benchmark for 3D human activity understanding. IEEE Trans. Pattern Anal. Mach. Intell. **42**(10), 2684–2701 (2020)
13. A. Yao, J. Gall, G. Fanelli, and L. J. Van Gool, 2011 "Does human action recognition benefit from pose estimation?"*British Machine Vision Conference*, Dundee, Scotland, pp. 67.1–67.11,
14. Z. Ge, S. Liu, F. Wang, et al, 2021 "YOLOX: Exceeding YOLO series in 2021," *arXiv preprint* arXiv:2107.08430, pp. 1–7
15. Sadiq, M., Masood, S., Pal, O.: Fd-yolov5: a fuzzy image enhancement based robust object detection model for safety helmet detection. Int. J. Fuzzy Syst. **24**(5), 2600–2616 (2022)
16. Yu, Y., Si, X., Hu, C., Zhang, J.: A review of recurrent neural networks: LSTM cells and network architectures. Neural Comput. **31**(7), 1235–1270 (2019)
17. Shen, X., Ding, Y.: Human skeleton representation for 3D action recognition based on complex network coding and LSTM. J. Vis. Commun. Image Represent. **82**, 1–9 (2022)
18. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: "BERT: Pre-training of deep bidirectional transformers for language understanding", *Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4171–4186. Minneapolis, Minnesota (2019)

19. T. H. Le, T. M. Le, and T. A. Nguyen, 2023. "Action identification with fusion of BERT and 3DCNN for smart home systems," *Internet of Things*, pp. 1–20, vol. 22
20. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, 2015, "Learning spatiotemporal features with 3D convolutional networks," *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 4489–4497
21. K. Liu, W. Liu, C. Gan, M. Tan, and H. Ma, 2018 "T-C3D: Temporal convolutional 3D network for real-time action recognition," *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, pp. 7138–7145
22. V.-M. Khong and T.-H. Tran, 2018. "Improving human action recognition with two-stream 3D convolutional neural network," *IEEE 1st International Conference on Multimedia Analysis and Pattern Recognition (MAPR)*, Ho Chi Minh City, Vietnam, pp. 1–6
23. Z. Zhao, W. Zou, and J. Wang, 2020. "Action recognition based on C3D network and adaptive keyframe extraction," *IEEE 6th International Conference on Computer and Communications (ICCC)*, Chengdu, China, pp. 2441–2447
24. Du, Y., Fu, Y., Wang, L.: Representation learning of temporal dynamics for skeleton-based action recognition. IEEE Trans. Image Process. **25**(7), 3010–3022 (2016)
25. Zhang, S., Yang, Y., et al.: Fusing geometric features for skeleton-based action recognition using multilayer LSTM networks. IEEE Trans. Multimedia **20**(9), 2330–2343 (2018)
26. Zhang, P., Lan, C., Xing, J., et al.: View adaptive neural networks for high performance skeleton-based human action recognition. IEEE Trans. Pattern Anal. Mach. Intell. **41**(8), 1963–1978 (2019)
27. Tang, Y., Liu, X., Yu, X., et al.: Learning from temporal spatial cubism for cross-dataset skeleton-based action recognition. ACM Trans. Multimed. Comput. Commun. Appl. **18**(2), 1–24 (2022)
28. S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," *Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA, pp. 7444–7452, Apr. 2018.
29. M. Li, S. Chen, X. Chen, Y. Zhang, et al, 2019. "Actional-structural graph convolutional networks for skeleton-based action recognition," *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, USA, pp. 3590–3598
30. Zhang, X., Xu, C., Tian, X., Tao, D.: Graph edge convolutional neural networks for skeleton-based action recognition. IEEE Trans. Neural Netw. Learning Syst. **31**(8), 3047–3060 (2020)
31. Shi, L., Zhang, Y., et al.: Skeleton-based action recognition with multi-stream adaptive graph convolutional networks. IEEE Trans. Image Process. **29**, 9532–9545 (2020)
32. Yang, H., Yan, D., et al.: Feedback graph convolutional network for skeleton-based action recognition. IEEE Trans. Image Process. **31**, 164–175 (2021)
33. Huang, Z., Qin, Y., et al.: Motion-driven spatial and temporal adaptive high-resolution graph convolutional networks for skeleton-based action recognition. IEEE Trans. Circuits Syst. Video Technol. **33**(4), 1868–1883 (2023)
34. Liu, Y., Zhang, H., et al.: Skeleton-based human action recognition via large-kernel attention graph convolutional network. IEEE Trans. Visual Comput. Graphics **29**(5), 2575–2585 (2023)
35. Li, M., Chen, S., Chen, X., et al.: Symbiotic graph neural networks for 3D skeleton-based human action recognition and motion prediction. IEEE Trans. Pattern Anal. Mach. Intell. **44**(6), 3316–3333 (2022)
36. Lugaresi, C., Tang, J., et al.: "Mediapipe: A framework for perceiving and processing reality", *IEEE Computer Vision and Pattern Recognition (CVPR)*, pp. 1–4. Long Beach, California, USA (2019)
37. N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, 2019. "MARS: Motion-augmented rgb stream for action recognition," *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Long Beach, USA, pp. 7882–7891
38. B. Yang, X. Zhang, et al., 2024. "Fast Multiview Anchor-Graph Clustering," *IEEE Transactions on Neural Networks and Learning Systems*, early access
39. Yang, B., Wu, J., et al.: Discrete correntropy-based multi-view anchor-graph clustering. Inform. Fusion **103**, 102097 (2024)