Soccer Highlight Generator

Anirudh Ramesh Narayanan
Department of Computer Science
Golisano College of Computing and Information Sciences
Rochester Institute of Technology
Rochester, NY 14586
an9425@cs.rit.edu

***Abstract*—In the realm of sports, soccer stands as a global phenomenon, drawing millions of viewers each week. While live matches are popular, there is a growing demand for concise game highlights, catering to fans with time constraints or varied interests. Despite this demand, the soccer industry remains relatively underdeveloped in terms of automation. Be it automation in detecting event occurring in the game, keeping track of the statistics of the game or highlight generation. The current process of highlight generation predominantly relies on labor-intensive manual processes. This paper introduces an innovative methodology for identifying key in-game events using a sophisticated combination of a Video Classifier and Yolo object detection model. Uniquely, it introduces a distinctive approach by utilizing augmented data displayed by broadcasters, including visuals for bookings, substitutions and a live scoreboard to accurately monitor key events. This method contrasts with most current techniques, which predominantly rely on multi-modal video-audio or only video classification models for event identification. The work load of identifying the events is split up between the Video Classifier and the Yolo model based on the broadcast videos available in the dataset used in this paper. This approach aims to streamline the process of highlight reel production, transforming raw game footage into a succinct yet comprehensive narrative of the match. While the proposed system shows considerable promise in capturing essential moments of soccer games, it faces challenges in handling intricate player movements. The paper culminates with recommendations focusing on refining the model's precision and reliability. These recommendations include expanding the YOLO model's classification capabilities through the incorporation of additional classes to enhance categorical granularity. Secondly, employing a dataset of superior resolution would likely bolster the precision of scoreboard tracking. Lastly, the construction of a more voluminous and diverse dataset for the video classifier is advised to improve its robustness and accuracy in video analysis tasks.**

I. Introduction

Football, also known as soccer, captivates the hearts of billions globally, with fans flocking to stadiums every game-day or eagerly gathering around their TV screens to cheer on their beloved teams in games that ignite passion and excitement. The most recent FIFA World Cup final, a captivating contest between Argentina and France, gathered an estimated viewership of 1.5 billion fans worldwide, as reported by FIFA. This staggering viewership shows the immense global appeal of football, a sport whose popularity continues to ascend. Furthermore, while millions of fans tune in weekly to witness their beloved teams compete, a substantial number are compelled to miss out on the live action due to commitments such as work and other obligations. There are also individuals who may find the duration of a match too lengthy for their viewing preferences. These fans often rely on official game highlights produced by the broadcasters, allowing them to experience the excitement of the matches at their convenience.

This, in turn, opens up revenue streams for event organizers and official broadcasters, who can capitalize on this demand by distributing these highlights across platforms such as YouTube and various streaming services. This presents an opportunity for an innovative tool capable of analyzing game footage to create engaging highlights by assessing in-game events. Such a tool could revolutionize content creation, offering broadcasters significant time and cost savings by automating the highlight generation process, which traditionally requires manual effort.

This paper introduces an innovative approach to creating dynamic highlight reels in soccer by pinpointing six pivotal events. These events include :

1) Penalties
2) Free-Kicks
3) Corners
4) Yellow Cards
5) Red Cards
6) Substitutions

We have trained a Video Auto-Encoder (VideoMAE), specifically tailored to discern penalties, free-kicks, and corners within our dataset, which consists of 150 videos per category. The dataset is divided, dedicating 70% to training and 15% each to validation and testing. The dynamic broadcast nature of soccer, characterized by constant camera transitions to capture the most engaging angles, presents a classification challenge.

To mitigate this, our methodology deploys a multi-class image classifier that segments the video feed into three distinct parts:

1) A segment containing frames which cover the wide-angle overviews of the pitch.
2) A segment containing frames which cover the close-ups of the players, referees, coaches, and audience reactions.
3) A collection of videos representing the in-game replays of key moments as selected by the broadcasters.

Segmenting the broadcast video into distinct categories prior to employing a video classification model significantly enhances its accuracy. This method allows the model to more confidently learn and interpret the temporal changes occurring between frames, leading to more precise event detection.

In elite soccer leagues, broadcasters enhance live feeds with real-time updates like card alerts, substitutions, and dynamic scoreboards. We utilize a YOLOv8 model to exploit these features, offering a significant enhancement over conventional approaches that typically overlook such broadcast elements. While lower-tier soccer might lack these sophisticated augmentations, our approach is designed to excel in environments where such data is accessible, markedly improving event detection accuracy. This method stands to revolutionize event recognition in soccer, bridging the gap between high-end broadcasts and more modest productions. Our dataset features 250 images for substitutions and bookings (which is a consolidated class for both red and yellow cards) and 690 images for the scoreboard. By employing computer vision techniques and HSV color space analysis, we enhance our model's ability to discern between card types, thus improving the accuracy and quality of moment detection and highlight generation, especially when additional broadcaster information is accessible. This approach showcases a significant progression in video classification, serving as a complementary technique to pure video analysis.

For goal detection, we use Amazon Web Services (AWS) Rekognition to analyze the scoreboard, which is extracted from the Yolov8-detected region. AWS Rekognition identifies all text in this area, and a Python script is used to extract the score. To optimize computational efficiency, the scoreboard is updated every 10 seconds, ensuring accurate tracking of goals. By identifying these key moments, we can assemble an engaging and comprehensive highlight reel that captures the essence of the game.

To facilitate transparency and enable collaborative advancements, we have made our complete implementation available to the public. The code can be accessed at our GitHub repository [13].

II. Background

Event detection in soccer is a challenging task, primarily due to the dynamic nature of the game and varying camera angles. The academic community has extensively explored diverse methodologies to accurately identify key events for statistical analysis and automated highlight generation. Among the machine learning techniques employed, YOLO object detection models have played a significant role. The research in paper [1] delves into evaluating the effectiveness of Faster RCNN models utilizing VGG16 and ResNet50 architectures, alongside YOLOv5. Despite the dataset's limitation to 231 images covering Corners, Free-kicks, Goals, and Fouls, the study revealed that the RCNN model with ResNet50 architecture exhibited superior performance, achieving an impressive accuracy rate of 95%. The VGG16-based model followed closely with a 92% accuracy. In stark contrast, the YOLO model demonstrated a markedly lower accuracy, standing at 30%.

In the paper [2], the authors introduce an innovative approach by integrating both audio and visual descriptors to analyze soccer games. This method stands out from traditional models that primarily focus on visual information. By employing audio cues, a second layer of validation is added, enhancing the reliability of event detection. The approach is multifaceted, involving the detection of referees' whistles, player movements, various facial expressions (players on the field, substitutes, and spectators), and specific broadcast cues like replay logos. Each of these components plays a role in identifying key moments. The methodology extends beyond simple event spotting; it is engineered to stitch together these moments into a cohesive highlight reel. A notable achievement of this method is its proficiency in accurately identifying goals, a crucial aspect of soccer highlights, with a success rate exceeding 70%. This dual-modality approach opens new avenues in sports analytics, demonstrating the potential for more sophisticated and accurate event classification systems in sports.

A primary challenge in sports event identification is the dynamic nature of the games and the visual ambiguity of different events from the broadcast camera's perspective. The technique proposed in paper [4] is particularly noteworthy. The authors describe using a YOLO model to pinpoint players on the field and deduce the homography matrix from the camera feed and sport-specific details. This matrix is used to transform the broadcast view into an overhead (bird-eye) pitch representation. Subsequently, player positions identified by the YOLO model are superimposed onto a standard sports field template, providing a clearer context for each event. This methodology proposed in paper [4] has been specified and implemented for the game of basketball. The authors of the paper [3] are inspired by paper [4], to implement the approach mentioned for the game of soccer. The methodology described in this paper uses a Camera Calibration for Broadcast Videos (CCBV). The CCBV framework is built on a three-stage process, each facilitated by a distinct neural network. Initially, the playing field is divided into zones using a U-Net architecture, with each zone defined by the surrounding field lines. Following this, an initial homography, or a projection from the field plane to the image, is established. Finally, a Siamese network is employed to convert both the segmented zones and pre-determined field templates into corresponding feature vectors for analysis. Player localization offers an enhanced perspective for training machine learning models to discern various soccer events. It proves especially beneficial

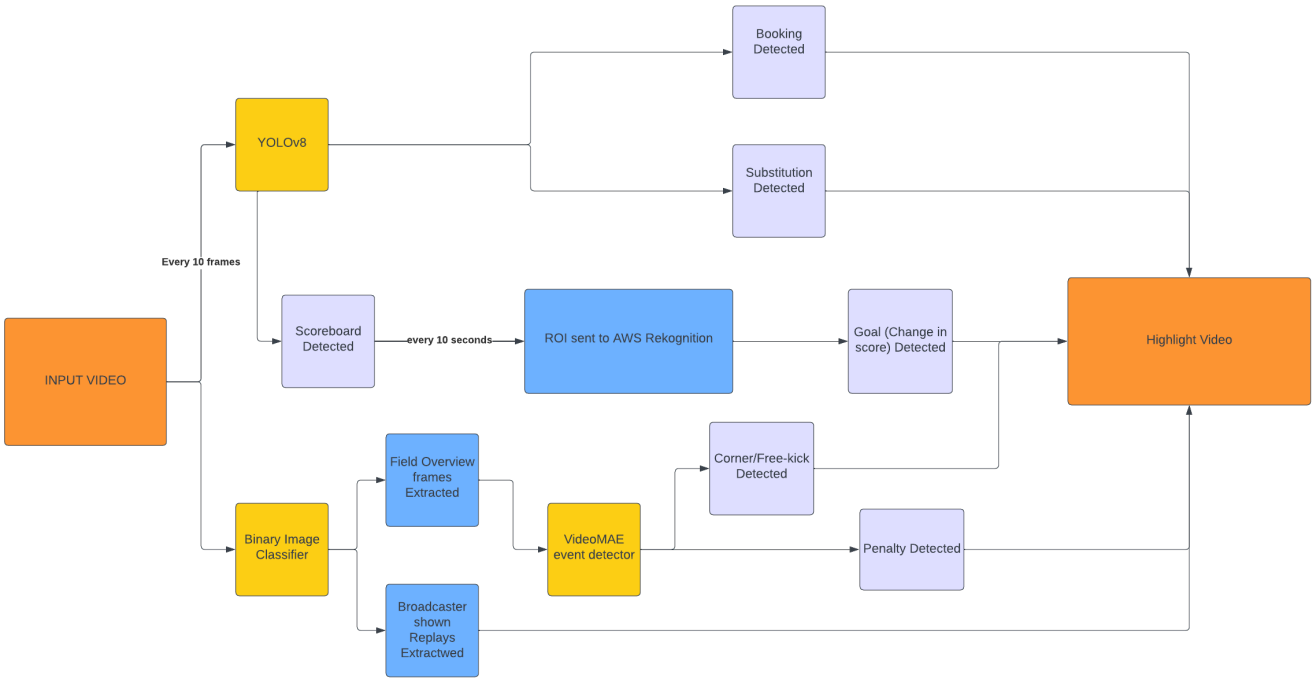


Fig. 1. Highlight Generation Flowchart

in distinguishing between events such as corners and free-kicks, where player clusters typically converge in the penalty area. The subtle distinction often lies in the position of the player initiating the play. Leveraging this method allows for the clear identification and classification of such events, even when visual cues are slight and not easily distinguishable. The authors of the paper use this information along with original dataset features to train a Context-Aware Loss Function (CALF) architecture to identify the different events in the game. The CALF architecture of the following layer: a frame feature extractor, a temporal segmentation module, and an action spotting module. When this model is trained it achieves a minimum precision of 32% on the class shot on target and a maximum precision of 76% on the class of corner. These results out perform all the previous accuracy achieved by other methodologies on this dataset.

III. Dataset

For this project, we utilized the open-source SoccerNetV2 dataset provided in the Paper [12], which is designed to foster contributions towards the automation of football analytics. The dataset consists of video footage of 550 games from the top football leagues in the world. These leagues include :

1) Premier-League (England)
2) La-Liga (Spain)
3) Champions-League (Europe)
4) Serie-A (Italy)
5) Ligue-1 (France)
6) Bundesliga (Germany)

The dataset offers video footage in two resolutions: low definition at 240p and high definition at 1024p at 25 frames per second(fps). The dataset offers 550 games for both the resolutions. Given the constraints of our hardware resources, this project employs videos at the low-definition 240p resolution to ensure efficient processing and analysis. The dataset also provides annotated labels corresponding to each event occurring in the game as shown in Table I.

TABLE I
Event Annotations from the Dataset

Game Time	Event Label	Time (ms)	Team	Visibility
1 - 00:00	Kick-off	0	Away	Visible
1 - 02:13	Ball out of play	113295	Not applicable	Visible
2 - 02:29	Throw-in	149168	Away	Visible

In Table I, the ”Game Time” column indicates the half of the match and the corresponding time in that half, i.e. as we can see in the second row, the values ’1 - 02:13’ indicates 2 minutes and 13 seconds have passed into the 1st half of the game and in the third row, the values ’2 - 02:29’ indicates that 2 minutes and 29 seconds have passed on the 2nd half of the game. The ”Event Label” column specifies the type of event occurring at that moment. The dataset offers annotated labels for 17 events in the game of football including events such as Free-Kicks, Penalty, Goal, Corner, etc. The ”Time (ms)” column shows how much time has passed in the half of the game in milliseconds. This information along with the fps(frames per seconds) information was used to extract exact

(a) Booking (b) Celebration

(c) Corner (d) Free-kick

(e) Goal (f) Open-Play

(g) Offside (h) Transition

(i) Penalty (j) Substitution

Fig. 2. Assortment of Game Scenes Used for Training the Model

frames from the video footage. The "Team" column identifies the team involved in the annotated event. The values for this column include 'Home', 'Away' and 'Not Applicable', where 'Home' refers to the team playing their home stadium, 'Away' refers to the visiting team and in certain events where neither team are involved in the event, the columns contains the value 'Not Applicable' as seen in the second row of Table I. The "Visibility" column clarifies whether the event is directly observable in the broadcast video or has been inferred by the pattern of the play in the game.

The Figure 2 contains some sample extracts from the dataset that was used for this project.

IV. Model Descriptions

A. Convolutional LSTM Model

The Convolutional Long-Short term memory model is predominantly used for classifying sequential data. The model used in this paper consisted of an architecture containing the following layers :

1) Input layer
2) ConvLSTM layer
3) MaxPooling layer
4) TimeDistributed Layer

As shown in the model architecture in Figure 3. The Input layer accepts an input of the shape [None, 20, 64, 64, 3] where 'None' represents in batch size which varies, '20' refers to the number of frames or the sequence length that the model will process for any given input video. '3' refers to the number of channels in a given input frame which in this case refers to the 'RGB'(Red, Blue, Green) channels in an image. The ConvLSTM layer, which is the core of the model, is designed to process spatial-temporal data by integrating the spatial feature learning capabilities of convolutional neural networks with the temporal learning abilities of LSTM networks. Within the ConvLSTM layer, the convolutional operations are embedded within the LSTM's gating mechanisms, allowing it to simultaneously learn spatial features and temporal dynamics from the data. The MaxPooling3D layer is used to condense the information and prevent overfitting and maintain computational load at a manageable level in the deeper layers of the model.

B. LCRN Model

The Long-term Recurrent Convolutional Network (LRCN) was selected for its effectiveness in handling sequential video data. An LRCN combines the spatial feature extraction capabilities of Convolutional Neural Networks (CNNs) with the temporal data processing abilities of Long Short-Term Memory networks (LSTMs). The architecture of the LCRN model comprises several key layers: an Input layer to receive the sequential data; multiple TimeDistributed Conv2D layers, which represent the CNN components for spatial feature extraction, TimeDistributed MaxPooling layers to reduce dimensionality, TimeDistributed Dropout layers to reduce chances of overfitting, an LSTM layer for temporal dynamic learning and a final Dense layer to output the classification results.

As shown in the model architecture in Figure 4. The Input layer is the same as used for the LSTM layer consisted of sequence length, input frame dimensions and the number of channels in the input frame. After the input layer, there is a series of CNN layers, each followed by a MaxPooling layer and dropout Layer, the CNN layers extract the spatial features from the input, while the Maxpooling layer is used to condense the information and the dropout layer is used to prevent overfitting and increase the generalization capability of the model. Finally, an LSTM layer is used to learn the temporal or sequential information in the input and the dense layer or the output layer predicts the class.

C. Video Mask AutoEncoders

This model has been fine-tuned to detects tasks in soccer based on the model provided in paper [5]. The paper shows how masked autoencoders are efficient in learning sequential

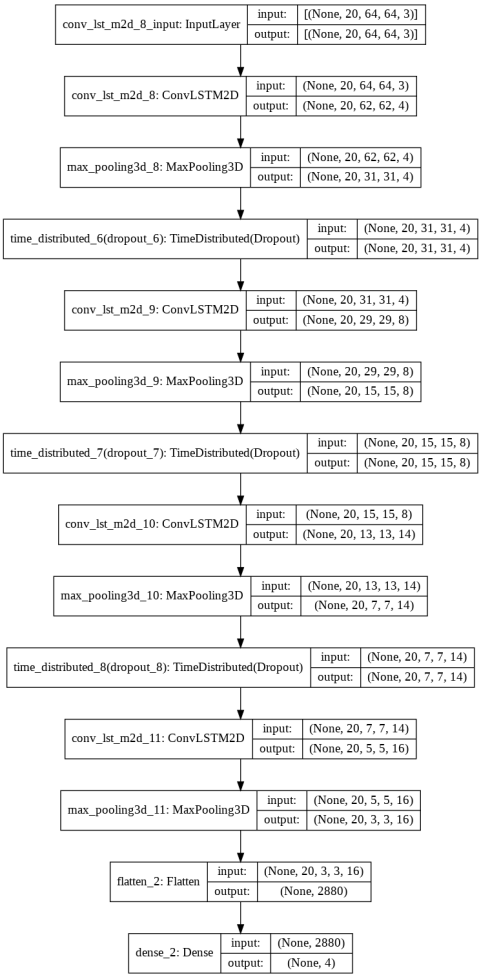


Fig. 3. Convolutional LSTM Model Architecture

data. The approach adopted in the paper, includes masking of random patches in the input image and then to reconstruct these missing pixels. The design of this model consists of two core designs that include an encoder-decoder architecture along with a secondary lightweight decoder that reconstructs the original image using mask tokens. The paper shows that this idea is efficient for developing models that are scalable and can achieve good generalization. The papers also suggests using a higher masking ratio (approximately 75%), which is a lot of higher than masking ratios generally used in Computer vision models (20%-50%). It was decided to fine-tune this model to detect the different events in soccer and hopefully achieve high accuracy and a usable model to create highlight of a soccer game. The specific model used in this paper was trained on the Kinetics-400 dataset for 1600 epochs. The Kinetics-400 dataset consists of 400 classes such as head-banging, salsa dancing, stretching leg, robot dancing, tickling, etc. The dataset consists of at least 400 videos of each class with an average length of 10 seconds per video. The model achieved a top-1 classification accuracy of 80.9% and a top-5 classification accuracy of 94.7% on the test set of the

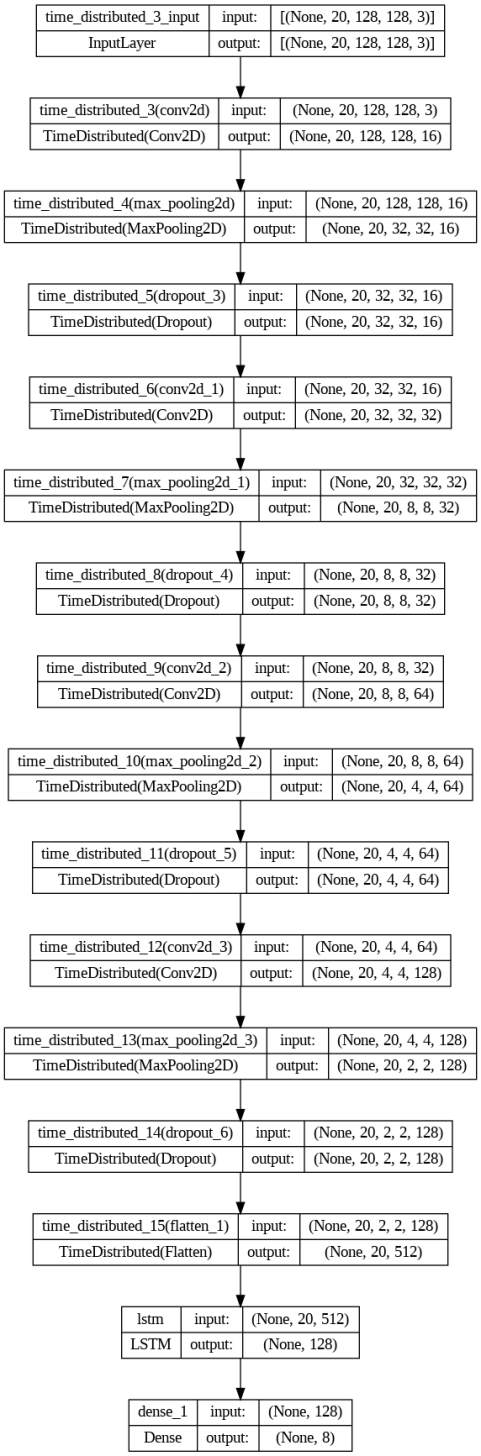


Fig. 4. LCRN Model Architecture

Kinetics-400 dataset. Based on these results, the model seemed promising for the task required.

D. YOLOv8

YOLO (You only look once) models are well-known for their speed and efficiency in real-time object detection. YOLOv8 is the latest iteration of the model as the time

of writing this paper, which provides better accuracy and increased speed than it's predecessors.

> The YOLOv8 architecture consists of a convolutional neural network divided into the backbone and the head. The backbone is a modified CSPDarknet53 with 53 convolutional layers, using cross-stage partial connections. The head employs multiple convolutional layers followed by fully connected layers for prediction tasks, and a self-attention mechanism to focus on relevant image features [6].

Based on these, the YOLOv8 model was used to detected certain events in the broadcast video.

E. Multi-Class Image Classifier

The model was initially proposed in paper [7]. The paper is inspired by the use of transformer models as a general purpose backbone model in the field of natural language processing (NLP) and proposes a similar backbone model that can be used in the field of computer vision. The model uses a shifting window approach in which the model initial starts off with small patches from the input and then combines them in the further layers. In each layer, as the partition window is shifted, it provides connections between the windows and corresponding pixels. This approach has a lower latency than the sliding window approach while maintaining model performance. This model was trained to classify an input image into one of three classes as further explained in section VI-A

V. Training

A. Image Classification

The initial methodology used for event classification in soccer games involved employing a ResNet-50 convolutional neural network to analyze and categorize game occurrences based on frames from the game clips. This process involved extracting individual frames from video footage representing the distinct event categories and subsequently training the ResNet-50 model on these categorized frames for event recognition. The dataset used to train the ResNet-50 model included a total of 2040 images, which was split into 17 classes, reaching an approximate 120 images per class. An example of the few of the classes are as shown in Figure 2. However, the results obtained from this strategy did not meet the anticipated performance benchmarks and the model could learn effectively in classifying the events solely based on a single frame of information. Consequently, it was decided to proceed with an alternative strategy involving video classification.

B. Video Classification

This approach employed video classification to enhance the accuracy of our event classification pipeline, seeking a more robust system for identifying key moments within the game. The process involved extracting video footage for each event following a similar approach as mentioned for the image classification approach. To refine the training of video classification models, the focus was narrowed to key events in a football match that are most relevant for generating highlight reels. These events included

1) Penalty Kicks
2) Goals
3) Free-Kicks
4) Corners
5) Bookings (Red and Yellow cards)
6) Offside
7) Substitution

The broadcast video also includes a transitional frame that is used to alternate between the live game and showing highlights of certain high intensity events that have occurred in the game (such as goals, a player injury, etc) as shown in Figure 2.h.

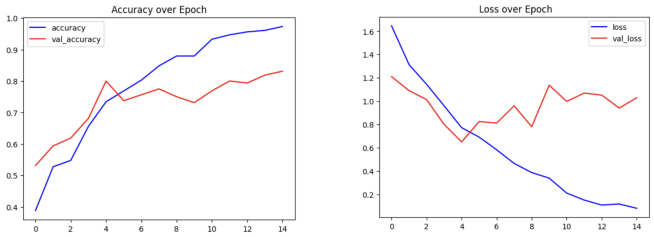
The initial method focused on identifying the nine events depicted in Figure 2, with the exception of open-play. The model demonstrated better event detection accuracy compared to a standard image classification model. However, it struggled to differentiate between free-kicks and corners due to their visual similarities, such as the clustering of players around the penalty box and the motion of play in the game. To enhance performance, these two classes were combined into a new class called Set-Piece. This consolidation is justified as the distinction between these events has a negligible impact on the quality of game highlights generated. It was also decided to introduce a new category named "Open-Play", encompassing segments of the game devoid of the specified events. This addition aimed to bolster the model's discernment abilities, particularly in scenarios where event identification is challenging. The models were then trained on this dataset consisting of these 9 events :

1) Transition
2) Set-Piece
3) Offside
4) Card
5) Goals
6) Penalty
7) Open-Play
8) Celebration
9) Substitution

The dataset for each class consisted of 150 videos on average across all the classes, ranging between 3-10 seconds in length.

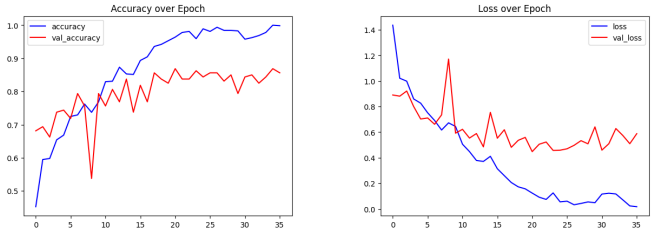
The metrics used to evaluate the model consisted of accuracy and loss across the training and validation sets. Accuracy was selected for its relevance to real-world applicability, offering a clear indication of the model's operational performance. Loss was tracked to gauge the model's prediction confidence, a crucial aspect of its reliability. Combined together, these metrics would provide an effective way to evaluate the model.

a) Convolutional LSTM Model: This model was trained initially for 100 epochs and the model showed signs of overfitting around the 15th epoch. In order to tackle this issue and improve the generalization capability of the model, an early stopping mechanism was added to the model training. In our implementation, an early stopping callback is instantiated to monitor the validation loss with a patience of 10 epochs.



(a) Accuracy (b) Loss

Fig. 5. LSTM Model training results



(a) Accuracy (b) Loss

Fig. 6. LCRN Model training results

This helped save computational cost by stopping the training once the model stops learning while simultaneously helping to prevent overfitting.

The results of the training are as shown in the Figure 5. As seen in Figure 5.a, the training and validation accuracy show a positive trend which can help us understand that the model is learning to classify the dataset, however Figure 5.b clearly indicate signs of overfitting as the validation loss plateaus around the 7th epoch and does not show improvement from that point forward. Different techniques such as dropout layers, L2 regularization were used to try to improve the model generalization capability, however the results did not show significant improvement to justify using the model as the event detector in the project pipeline.

b) LCRN Model: The training process of the LCRN was similar to the LSTM model,the model was trained for 200 epochs and it also included an early stopping mechanism monitoring the validation loss to prevent overfitting on the training dataset. For the LCRN model, the early stopping mechanism had a patience of 15 epochs on the validation loss.

The results of the training are as shown in the Figure 6. As we can see the results, the model performed similarly to the LSTM model in training as while the accuracy showed an increase, the loss plateaued around the 15th epoch. The model performance in training did not increase even after implementing different regularization techniques. Based on this information, it was decided this model does not possess sufficient capability to classify events for the required task.

c) Classification Result Comparison: The accuracy of the both the LSTM and the LCRN model for each class in the dataset as shown in Table II. As we can see, the LCRN model performed better on the test dataset than the LSTM model and could classify events more accurately. However there are certain events such as open-play, set-piece, where even the LCRN model failed to show promising classification results. In pursuit of a more accurate model for event classification within soccer games, the decision was made to refine a video classification model pre-trained on an extensive dataset, hoping to achieve better results.

TABLE II
COMPARISON OF LSTM AND LCRN MODEL ACCURACY

Class Name	LSTM Accuracy (%)	LCRN Accuracy (%)
Transition	83.3	85.2
Set-piece	21	57.1
Offside	50	83.3
Card	66.6	66.6
Goals	76.5	88.2
Penalty	85.9	92.9
Open-Play	70	60
Substitution	45.1	77.4
Celebration	53.7	72.2
Average	72.86	82.5

d) Video Mask AutoEncoders (MAE) - 1st Iteration: The training process for this model included fine-tuning the model mentioned section IV-C that has been extensively trained on large dataset for video classification. The initial training of the model across nine classes yielded encouraging accuracy, as indicated in Table III. Nonetheless, when applied to a three-minute video segment, the model produced numerous misclassifications. Further analysis suggested that discrepancies between the homogeneity of training data and the heterogeneity of actual game footage contributed to this issue. Training samples were sourced from single-camera perspectives, while live broadcasts involve switching between multiple cameras, introducing variations not accounted for in the training phase. Changes made to the model's training process are detailed in the following section of this report.

TABLE III
CLASS-WISE ACCURACY VIDEOMAE MODEL SECTION V-B0D

Class	Accuracy (%)
Card	88.89
Celebration	83.33
Goals	82.61
Offside	94.12
Open-Play	97.37
Penalty	95.83
Set-piece	94.64
Substitution	81.82
Transition	100.00
Average	92

(a) Substitution

(b) Booking

Fig. 7. Broadcaster shown events

(a) Input Image

(b) YOLO output

Fig. 8. Booking

(a) Input Image

(b) YOLO output

Fig. 9. Substitution

VI. Methodical Improvements for Better Accuracy Outcomes

A. Multi-Class Image Classifier

In order to tackle the problems mentioned in section V-B0d, a few modifications were implemented to the video classification process. It was decided to add a multi-class image classifier. The image classifier classifies an input frame into one of the following class

1) Close-up frame.
2) Overview frame.
3) Transition frame.

The ”Close-Up” frame describes the closeup shots of the players, referee and fans, the ”Overview” frame consists of a wide view of the pitch. The ”Transition” class includes frames which consists of a logo transition used by the broadcasters to show replays within the game. The training dataset for this model consisted of 500 frames of each class. An example of the ”Overview” is as shown in Figure 2.f, while an example of the ”Close-Up” is as shown in Figure 2.g. The Figure 2.h shows an example of the ”Transition” class. The model was trained for 10 epochs and the model achieves an accuracy of 97.4% in classifying the events in the testing set. Then a pipeline was created to use the model to separate the input video into separate videos, 1 video consisting of only the close-up shots, one video consisting of only the overview shots and another set of videos each for a replay shown by the broadcasters.

TABLE IV
Class-wise precision - YOLO

Class	Precision (%)
Booking (Card)	89.7
Logo	99.30
Scoreboards	98.14
Substitution	88.9
Average	91.2

B. YOLOv8

The soccer broadcast videos also contain instances where certain events of the soccer game are shown in broadcast videos. The Yolov8 model was decided to detect these events as they occur in the game of the football and then highlighted by the broadcasters. As we can in Figure 7.a which shows how broadcasters shows the event of a substitution and the Figure 7.b shows how broadcasters showcase the event of a booking. The broadcast video also contains a constant scoreboard which shows the current score between in the two teams at any given point in the game. It should be noted that each broadcaster has their unique way of showcasing these events as and when they occur they occur in a game.

For the training of this model, a total of 765 images were annotated with labels, the class booking and substitution had approximately 200 images with annotations, while the class scoreboard has 691 images with annotations. Along with these, another class logo was used to reduce the false positives produced by the model. The class logo refers to the logo of the league or the broadcaster logo shown in the video, which the model was sometimes confusing with the other classes. The images for each class were equally split among the 6 leagues provided in the dataset. The dataset was divided into three subsets: 70% for training, 15% for validation, and 15% for testing purposes.

The model was trained for 200 epochs and achieved an average precision of 91.2% and an average recall of 92.5% on the test set. The class-wise precision shown in Table IV illustrates that the model excels in recognizing scoreboard instances with impressive precision. In Tables V and VI-B, we can see the comparison between the precision values of the our YOLOv8 model with other models used to detect the respective events. It is important to note that the models mentioned in the comparison tables predict the events based on the player actions and positions, while our YOLOv8 model predicts these actions based on the information displayed by the broadcaster in the video. The comparative analysis demonstrates the superiority of our proposed method, which utilizes broadcaster displayed information, for detecting substitution and booking events over traditional video classification or action recognition models. This advantage is particularly remarkable given the notably smaller size of the dataset used

for our method, highlighting its efficiency and robustness in scenarios with limited data availability. Visual confirmation of the model's predictions can be seen in Figures 8 and 9, where the bounding boxes accurately reflect the anticipated events.

TABLE V
SUBSTITUTION EVENT DETECTION COMPARISON

Model	Training Images	Testing Images	Precision (%)
CALF-60-s [10]	1708	562	60.07
CALF-60-20 [10]	1708	562	46.70
CALF-120-40 [10]	1708	562	67.35
YOLOv8 (This paper)	180	35	88.9

TABLE VI
BOOKING EVENT DETECTION COMPARISON

Model Name	Training Images	Testing Images	Precision (%)
B-CNN [8] [11]	5500	500	66.86
OSME + MAMC using Res-Net50 [9] [11]	5500	500	61.70
OSME + MAMC using EfficientNetB0 [8] [11]	5500	500	79.90
YOLOv8 (This paper)	170	34	89.7

C. Video Mask AutoEncoders (MAE) - 2nd Iteration

Based on the YOLO model's success in identifying the booking and substitution events, while also identifying the scoreboard accurately, it was decided to re-train the Video classification model to identify the remaining key events in a football game. The new dataset consisted of the following events :

1) Transition
2) Set-Piece
3) Penalty
4) Open-Play

The remaining events/classes that were removed from the previous iteration included cards and substitution which is now detected using the YOLO model, while the other events removed includes goals and celebration which can be detected by keeping track of the scoreboard detected in YOLO model.

The dataset comprised around 150 videos for each category. After training across 40 epochs, the model reached an impressive Average accuracy of 95.34%. This high accuracy rate, evidenced by class-specific metrics as shown in table VII, suggests strong predictive performance.

TABLE VII
CLASS-WISE ACCURACY FOR VIDEOMAE SECTION VI-C

Class	Accuracy (%)
Open-Play	98.70
Penalty	91.67
Set-Piece	92.86
Transition	100.00
Average	95.34

VII. POST-PROCESSING AND HIGHLIGHTS GENERATION

A. Multi-Class Classifier Usage

The multi-class classification system processes each input frame, categorizing it into one of three predefined classes: Closeup, Overview, or Transition. This system employs a flag to monitor the current state of the video, distinguishing between live and replay modes. Depending on the classified category of the frame, it is accordingly appended to either the Closeup or Overview video segments. The detection of a Transition frame triggers a change in the system's state: the flag is set to true, signifying the commencement of a highlight sequence. Subsequent frames are then added to the highlight reel until the system identifies another Transition frame. Upon this detection, the flag reverts to false, indicating the end of the highlight sequence, and the process resumes from the beginning. To ensure that the same transition logo is not repeatedly trigger the flag, a cooldown timer of 1.5 seconds, equivalent to 37 frames, is implemented. This timer is activated once a transition frame is identified, effectively skipping over subsequent frames during this interval to prevent redundant toggling.

This process of separating live video footage from the in-game replays helps in avoiding adding duplicates video sequences to the final highlight reel created.

B. VideoMAE classifier

The 'overview' video segment is extracted employing the multi-class classifier, as elaborated in section VII-A. Following this, the VideoMAE classifier, which has undergone training as detailed in section VI-C, is utilized to identify and discern events within the video. To rigorously assess the classifier's accuracy, the overview video was segmented into several clips of varying durations. Specifically, we tested the classifier's performance on video segments with lengths of 3, 5, 7, and 10 seconds. Considering the outcomes garnered, the decision was made to segment the video into discrete intervals of 10 seconds in the final pipeline. In order to further reduce false positives, a cooldown timer of 20 seconds is implemented in order to suspend any event detection in the following time frame once a event is detected.

C. YOLOv8

The input video is sent to the YOLOv8 model, where it processes one in 10 frames to look for the events of

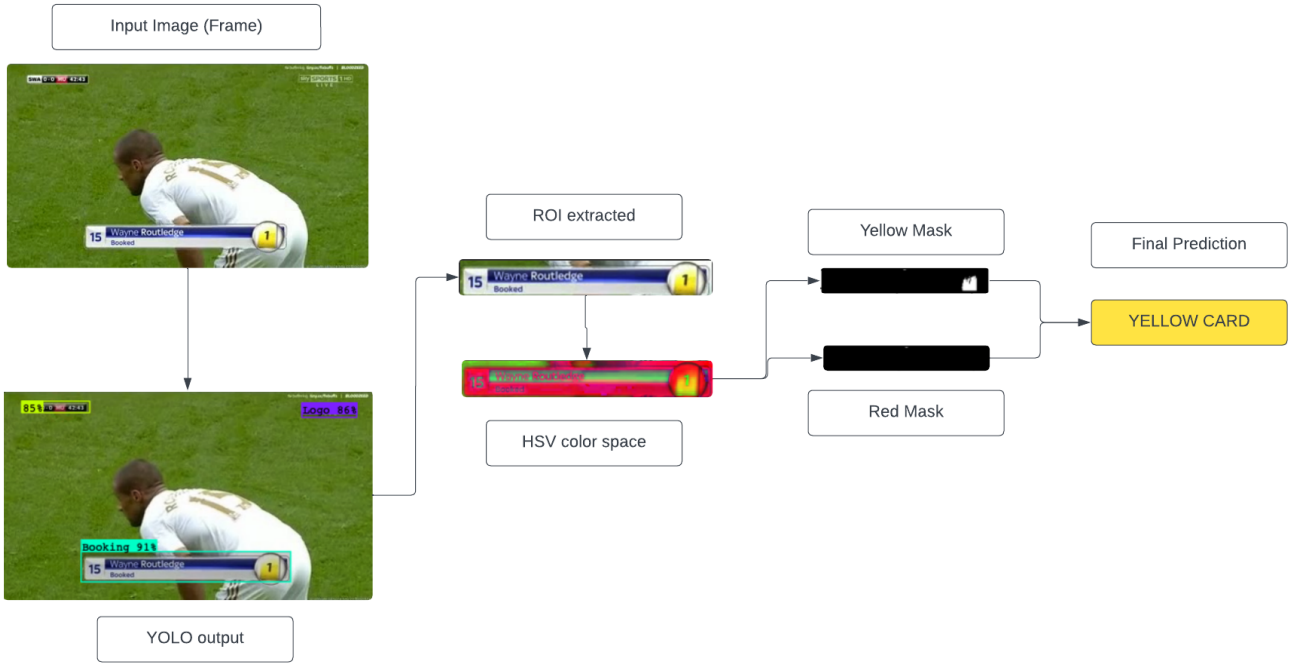


Fig. 10. Yellow Card Detection

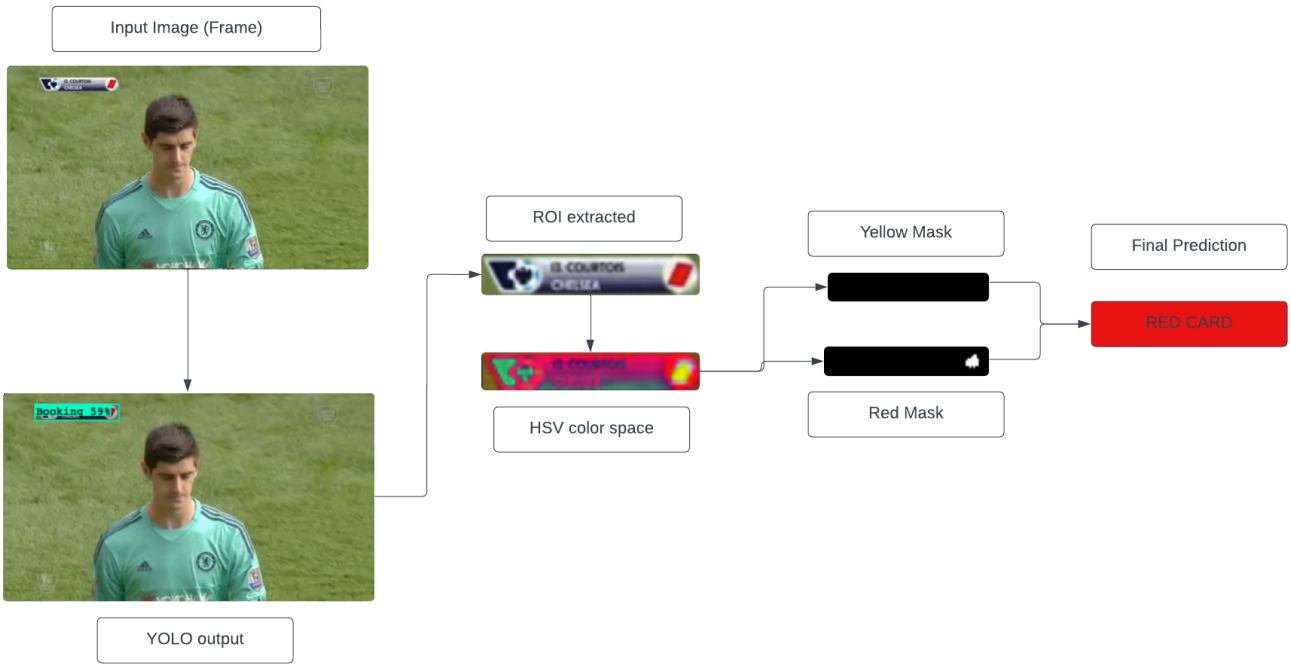


Fig. 11. Red Card Detection

Substitution and Booking, the scoreboard tracker is updated every 10 seconds, i.e. once for every 250 frames. The detailed process for both the event classifications in mentioned in the following sections.

1) Card Color Detection: Since Red Card is not a frequently occurring event in the game of football, the dataset did not consist of sufficient examples of the event, based on this, the decision was made to combine Red and Yellow card

events for the YOLO model and later use computer vision techniques to determine the color of the card.

Once the event Booking is detected by the YOLO model, a region of interest(ROI) in extracted from the frame where the event was detected using the bounding box coordinates returned by the model. The model returns the following information regarding the bounding box, 'x', 'y', 'width', 'height' where 'x' and 'y' denote the center of the bounding box. Once the ROI has been extracted, it is converted from the RGB(Red, Blue, Green) color space to the HSV(Hue, Saturation, Value) color space. This conversion has the following benefits, it is easier to define a color range for color segmentation in the HSV space where it can be done by defining a range of hues, while in the RGB space, it requires a relation between the 3 color channels. The HSV color space also separates the color information(hue) from the lighting information(value), which makes the process of identifying colors under different lighting conditions easier.

Once the ROI has been converted to the HSV color space, 2 masks (one each for red and yellow) are created using pre-defined values for each color. The values used for this project can be seen in Table VIII. The Hue component is measured in degrees from 0 to 179, representing the type of color. Saturation and Value are measured on a scale from 0 to 255, representing the intensity and brightness of the color, respectively. After creating this mask, we count the number of non-zero (non-black) pixels in each mask, and choose the color of the card as the color of the mask with more non-zero pixels. The process can be seen in Figures 10 and 11.

TABLE VIII
HSV COLOR RANGES FOR CARD COLOR DETECTION

Color	Hue (H)	Saturation (S)	Value (V)
Yellow (Lower Bound)	15	100	100
Yellow (Upper Bound)	30	255	255
Red (Lower Bound)	0	100	100
Red (Upper Bound)	20	255	255

2) Tracking Goals: To monitor goals during a game, it was determined that tracking changes on the scoreboard would serve as a reliable indicator of scoring events. This methodology includes the steps of capturing the Region Of Interest (ROI)—the scoreboard—every 10 seconds from the video feed. Subsequently, Amazon Web Services (AWS) Rekognition's Text in Image feature is employed to decipher the text within the ROI, thereby maintaining an accurate score tally. For AWS Rekognition to function effectively, both the image and the ROI are required inputs for text detection during the API call. This procedure is illustrated in Figure 12.

VIII. EVALUATING THE PERFORMANCE OF THE FINE-TUNED VIDEOMAE MODEL ON LOWER-RESOLUTION VIDEOS

The dataset used for the 240p resolution videos is of the size 971.2Mb, indicative of richer detail and higher quality,

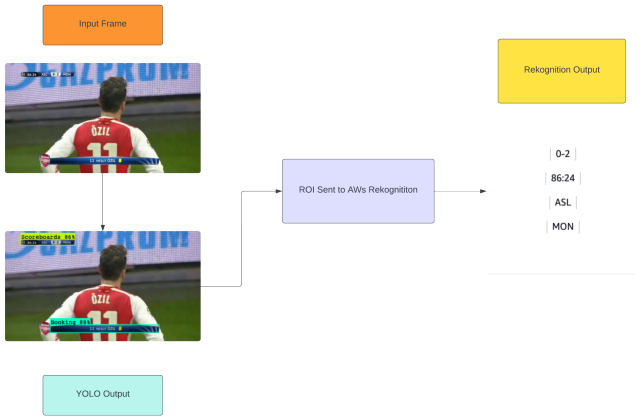


Fig. 12. Scoreboard tracking Using AWS Rekognition

whereas the dataset for the 144p resolution videos stands at a significantly reduced size of 97.5Mb. The stark decrease in file size correlates with a loss of visual information, which is paramount for the model's performance.The analysis of the VideoMAE model's performance across varying video resolutions, as shown in Table IX, reveals significant variations in accuracy, underscoring the challenges posed by lower-resolution inputs. Notably, when processing 144p resolution videos, the model demonstrates a marked decrease in efficacy. The accuracy in discerning 'Transition' events, although still relatively high at 96.09%, contrasts sharply with the substantial drop observed in 'Open-Play' events, plummeting from a robust 98% to a mere 71.21%. This decline is also reflected in the Average accuracy, which falls from 95.34% for 240p videos to 78.97% for 144p videos.

This decrease in accuracy, particularly evident in the finer delineation of 'Open-Play' events, emphasizes the crucial role resolution plays in the model's capacity to accurately classify video content. The diminution in detail and clarity in 144p videos likely impairs the model's ability to capture and analyze subtler visual cues, leading to reduced performance. This observation highlights a critical trade-off between computational efficiency and classification accuracy, especially pertinent in video classification tasks where input resolution varies.

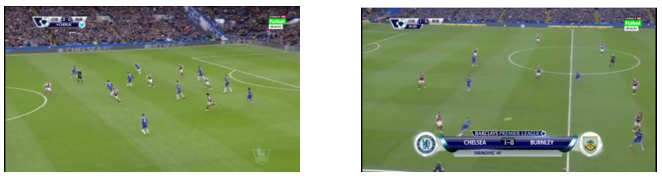
TABLE IX
ACCURACY COMPARISON BETWEEN 240P AND 144P RESOLUTION VIDEOS

Class	144p Accuracy (%)	240p Accuracy (%)
Setpiece	75.70	92.86
Transition	96.09	100.00
Penalty	80.37	91.67
Openplay	71.21	98.70
Average	78.97	95.34

IX. Discussion / Results

A. Highlight Generation Results

This section delves into the outcomes of the highlight generation process as delineated in section VII. The YOLOv8 model demonstrated proficiency in tracking the designated events within the video footage. Nonetheless, there were instances of false positives, exemplified in Figure 13. Notably, the model erroneously recognized the scoreboard as a booking in Figure 13.a, owing to the similarity between the scoreboard's display and the way broadcasters present booking information. Likewise, in Figure 13.b, the score update, appearing at the screen's center, was mistakenly identified as a booking event. It is important to note that while most of these false positives were associated with a lower confidence level (around 70%), which contrasts with the higher confidence levels (typically 85% to 95%) of accurate predictions, there were certain frames where the model exhibited equal or greater confidence in these incorrect identifications than in the correct ones. Consequently, the implementation of a mere confidence threshold was inadequate for filtering out these false positives.



(a) False Positive (b) False Positive

Fig. 13. YOLO False Positives

The multi-class classifier demonstrated remarkable efficacy in isolating in-game replays presented during live broadcasts. In a practical test involving 45 minutes of game footage, the model exhibited exceptional accuracy. It successfully identified and extracted all 17 replays showcased during the game, utilizing the transition phase as a key indicator. This level of precision in identifying each replay instance underscores the classifier's practical utility and reliability in real-world applications. Along with this the model could also effectively separate the input video of the game into 2 parts, a video consisting of only the close-up frames and another video only consisting of overview frames effectively with an accuracy of 97.5% using a confidence threshold of 98% which is used in order to remove any false positives.

The VideoMAE classifier underwent testing on the overview video, which was extracted using the multi-class classifier. It demonstrated a commendable precision rate of 66%. Specifically, the model accurately detected 7 set-piece events which includes free-kicks and corners. However, it also produced some false positives, as depicted in Figure 14. A notable example is shown in Figure 14.a, where the frame closely resembles the typical player positioning and clustering observed during a free-kick in soccer. This similarity provides context for the occurrence of this particular false positive and highlights the nuanced challenges in event detection within the game. The Table X refers to the comprehensive overview of various tasks within the project's pipeline, detailing the models used, the division of training and testing data, the type of model for each task, and the corresponding accuracy achieved.

(a) False Positive (b) False Positive

Fig. 14. VideoMAE False Positives

B. The Significance of Video Quality

The significance of video quality in the context of model performance is extensively discussed in Section VIII. This section details how the VideoMAE model's ability to classify events is influenced by the resolution of the input videos. The comparative analysis between 240p and 144p video resolutions underscores the pivotal role of video quality in achieving high accuracy in Machine learning-based event analysis.

C. Comparison Results between Official Highlights and different Pipelines tested

In order to test the performance of the methodology described in this paper, it was decided to compare the results of the pipeline with the official highlight uploaded on the internet. The game decided for this comparison was a premier league game from the Year 2014-15 between Chelsea and Burnley which ended with a scoreline of 1-1.

To evaluate the models' accuracy in generating highlights, two distinct pipelines were implemented. The primary process is depicted in the flowchart illustrated in Figure 1. Each pipeline functions as follows:

1) The first pipeline employs a multi-class classifier to separate the input video into two categories: 'overview' and 'closeup' videos, without extracting replay segments.
2) The second pipeline advances this process by extracting replay segments, in addition to dividing the input video into 'overview' and 'closeup' videos.

These approaches allow for a comparative analysis of the models' performance in different highlight extraction scenarios.

The official highlight reel comprised 2 minutes and 7 seconds of footage. In comparison, the highlight video generated by the first pipeline extended to 12 minutes and 20 seconds, while the second pipeline produced a slightly shorter compilation, totaling 10 minutes and 35 seconds. The information regarding the events extracted can be seen in the Table XI.

The original highlight video was concise, showcasing only critical moments such as goals, a pair of fouls, corners, a card, and one substitution. While this brief summary highlighted key events affecting the score, it omitted several other significant plays that could provide a fuller narrative of the match's proceedings. The first pipeline generated a highlight reel

TABLE X
SUMMARY OF ALL TASKS IN HIGHLIGHT PIPELINE

Task	Model	Training Data		Type of Model	Accuracy
		Train	Test		
Overview Frames	Swin-Binary classifier	550 Images	150 Images	Image Classification	97%
Close-up Frames	Swin-Binary classifier	500 Images	130 Images	Image Classification	95%
Transition (Logo) Frames	Swin-Binary classifier	540 Images	140 Images	Image Classification	99%
Booking detection	YOLOv8	175 Images	75 Images	Object Detection	89.7%
Substitution detection	YOLOv8	175 Images	75 Images	Object Detection	88.9%
Goals detection	YOLO and AWS Rekognition	550 Images	140 Images	Object Detection and Text Extraction	98.14%
Set-Piece detection	VideoMAE	140 Videos	40 Videos	Video Classification	92.86%
Penalty Detection	VideoMAE	120 Videos	35 Videos	Video Classification	91.67%

that included some repeated events, as it did not extract the broadcaster's replay segments. This resulted in the duplication of two corners and one free-kick within the highlight reel generated. The second pipeline refined this approach by using broadcaster replays as a foundation, augmenting them with events detected by the YOLOv8 model and the VideoMAE classifier to produce a comprehensive and succinct highlight reel.

TABLE XI
HIGHLIGHT VIDEO COMPARISON

Video Information	Original Highlight	1st Pipeline	2nd Pipeline
Highlight Length	2:07	12:20	10:35
Goal	2	2	2
Foul	2	0	2
Free-Kick	0	4	3
Corner	2	7	5
Card	1	5	5
Substitution	1	4	4
False-Positives / Repeat	0	12	6
Attacks	0	1	18

Both the pipelines could effectively keep track of the goals, substitution and booking events, however there were a few false positives that were detected by the methodology as discussed in section IX-A. The first pipeline registered 12 false positives, while the second pipeline identified 6. These inaccuracies occurred during open play, where the positioning of players and the game's momentum bore resemblance to scenarios typically involving corners and free-kicks. One of the reasons for these additional false positives identified by the model in the 1st pipeline is due to the replay videos not being extracted.These replays typically capture high-intensity moments near the pitch's ends, where player clustering often leads to confusion for the model. Another challenge encountered by the model stemmed from the 240p resolution of the videos employed in this project. This led to numerous misclassifications while trying to keep track of the score by extracting the text from the scoreboard. The ensuing sections will discuss potential improvements to address these issue.

X. FUTURE WORK

A. YOLOv8

The YOLOv8 model trained for this project contained 695 images, with around 200 annotated images for the substitution and booking event, however this did not keep in mind the other information displayed by the broadcasters during the game. These events include:

1) Game hashtag shown in the start of the game.
2) Referee name display.
3) Huge scoreboard shown to indicate change in score.
4) Various advertisments shown by the broadcasters

Adding these classes to the dataset while training the YOLO model should result in better accuracy and reduce the false positives. This should provide an even more accurate YOLOv8 model which will be able to detect the events with good precision.

B. Score-Tracker

The methodology in this paper depended on Optical Character Recognition (OCR), in order to extract text from the scoreboard to detect the goals. An OCR algorithm depends on the quality of the input image, however the dataset in this project used videos with pixel resolution of 240p, this resulted in several misclassifications by the OCR algorithm. These results should show considerable improvement by using videos with higher resolution.

C. VideoMAE

The video classification model for this project was trained on a dataset of approximately 150 videos per class. This is a comparatively smaller size of dataset with respect to the task of identifying the different classes that occur in the game. The dataset also does not consider many classes that occur in the game, such as shot on target, shot off target, clearance, etc. Adding these events to the dataset while simultaneously increasing the size of the dataset for each class should result in better event detection accuracy.

XI. Conclusion

Integrating the YOLOv8 model with a fine-tuned, state-of-the-art video classifier pre-trained on an extensive dataset yielded promising results in creating soccer game highlights. While the methodology outlined in this paper succeeded in capturing most key events of the game, it's not without its limitations. As discussed in Section X, changes and improvements can be made to the system to improve the event detection and classification methodology which will result in creation of a better highlight reel.

Acknowledgment

The author would like to thank Dr. Leon Reznik and Mr. Sergei Chuprov for their invaluable support, insightful advice, and technical assistance to make this research possible.

References

[1] N. Darapaneni, P. Kumar, N. Malhotra, V. Sundaramurthy, A. Thakur, S. Chauhan, K.C. Thangeda, and A.R. Paduri, "Detecting Key Soccer Match Events to Create Highlights Using Computer Vision," *arXiv preprint arXiv:2204.02573*, 2022.

[2] A. Raventos, R. Quijada, L. Torres, and F. Tarrés, "Automatic Summarization of Soccer Highlights Using Audio-visual Descriptors," *SpringerPlus*, vol. 4, no. 1, pp. 1–19, 2015. DOI: 10.1186/s40064-015-1204-4.

[3] A. Cioppa, A. Deliege, F. Magera, S. Giancola, O. Barnich, B. Ghanem, and M. Van Droogenbroeck, "Camera Calibration and Player Localization in SoccerNet-v2 and Investigation of Their Representations for Action Spotting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2021, pp. 4537-4546.

[4] L. Sha, J. Hobbs, P. Felsen, X. Wei, P. Lucey, and S. Ganguly, "End-to-End Camera Calibration for Broadcast Videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[5] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick, "Masked autoencoders are scalable vision learners," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16000–16009, 2022.

[6] Understanding YOLOv8 Architecture, Applications & Features. Available: https://www.labellerr.com/blog/understanding-yolov8-architecture-applications-features.

[7] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012–10022, 2021.

[8] Mingxing Tan and Quoc Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," *International Conference on Machine Learning*, pp. 6105–6114, PMLR, 2019.

[9] Ming Sun, Yuchen Yuan, Feng Zhou, and Errui Ding, "Multi-attention multi-class constraint for fine-grained image recognition," *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 805–821, 2018.

[10] O.A. NergårdRongved, M. Stige, S.A. Hicks, V.L. Thambawita, C. Midoglu, E. Zouganeli, D. Johansen, M.A. Riegler, and P. Halvorsen, "Automated Event Detection and Classification in Soccer: The Potential of Using Multiple Modalities," *Machine Learning and Knowledge Extraction*, vol. 3, no. 4, pp. 1030–1054, 2021. DOI: 10.3390/make3040051.

[11] A. Karimi, R. Toosi, and M. A. Akhaee, "Soccer event detection using deep learning," *arXiv preprint arXiv:2102.04331*, 2021.

[12] Silvio Giancola, Mohieddine Amine, Tarek Dghaily, and Bernard Ghanem, "SoccerNet: A Scalable Dataset for Action Spotting in Soccer Videos," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[13] Anirudh Ramesh Narayanan. "Soccer Highlight Generator" 2023. Available: https://github.com/Anirudhrn98/SoccerHighlightGenerator.