LONG SHORT-TERM MEMORY MODEL-ENABLED HIGHLIGHT DETECTION IN SOCCER

WILLIAM QI [WILLQI@SEAS], KYLE WU [KKYW30@SEAS], EMMET YOUNG [EMMETY@SEAS]

ABSTRACT. This study describes a Long Short-Term Memory-enabled approach to accomplish the task of detecting highlights in soccer games. A deep neural network pre-trained on the Kinetics 600 dataset was used to generate token "predictions" for each five-second segment of these game videos. An LSTM model was subsequently trained on these tokenized predictions along with audio data to establish sequential relationships for predicting the presence of highlights. We report a training accuracy of over 87 percent along with an average subjective evaluation of 5.2 out of 7 across 10 evaluators and 3 of our generated highlight reels. Our work provides a promising approach to automated highlight detection and recap generation.

1. INTRODUCTION

Highlight detection in sports enables viewers to enjoy the most exciting moments of sports games, isolating key moments from matches that can be hours long. These key game moments provide easy engagement with the audience and are often integral parts of advertising and content production. Highlight detection is especially relevant for soccer, where key moments are relatively sparse despite long matches of uninterrupted play.

However, isolating highlights can be quite challenging. Varying broadcast angles, unimportant stoppages of play, and irregular movement patterns, among other nuisances, pose significant challenges to separating true highlights from merely unpredictable sequences. Furthermore, the notion of a "true" highlight is quite subjective, as many viewers may have differing opinions on what is most important in a video summary of a sports game. While most will consider important scores and exciting plays as highlights, some may be more inclined to include major injuries while others might argue that showing the crowd's excitement is important to the game. Our project aims to solve this problem and eliminate the inherent subjectivity via a deep learning approach. By leveraging recurrent architectures and various methods of anomaly detection, we hope to create our own model that can both identify noteworthy segments of a soccer game and systematically decide what constitutes a highlight.

1.1. **Contributions.**

- Designed multi-modal LSTM to synthesize video and audio data to generate highlight clips
- Proposed a novel algorithm to quantify surprise and significance of events in soccer match to more objectively evaluate highlight likelihood and model outputs
- Achieved over 87% classification accuracy in predicting token of next action across full dataset
- Attained average evaluation rating of 5.2 out of 7 for our generated highlights

2. BACKGROUND

Recurrent architectures provide us with a way to analyze sequences of data through time. In such architectures, a hidden state—dependent on all past data seen so far—along with the current datum constitute sufficient statistics that are sufficient for predicting the state in the next time step. The task of predicting the future state is a non-trivial task, especially when establishing long-term temporal dependencies, because backpropagation through time for recurrent architectures is prone to exploding and vanishing gradients.

One type of recurrent architecture is Long Short-Term Memory, which helps alleviate the vanishing gradient problem. It consists of an input, forget, and output gate to better control the flow of gradients. The gates are defined as:

$$
\begin{aligned}
i_{t+1} &= \sigma(w_{hi}h_t + w_{xi}x_{t+1}) \\
f_{t+1} &= \sigma(w_{hf}h_t) + w_{xf}x_{t+1} \\
o_{t+1} &= \sigma(w_{ho}h_t + w_{xo}x_{t+1})
\end{aligned}
$$

The forget gate decides which information to forget or keep from the hidden state and current input, the input gate determines which aspects of the data should be added to the memory cell, and the output gate determines what information becomes part of the LSTM's current output based on the updated hidden state. While not fully alleviating the gradient problem, the LSTM's gate-control mechanisms can handle considerably longer input sequences than the most basic recurrent neural networks.

Convolutional Neural Networks (CNNs) leverage convolutional layers to extract hierarchical features by applying localized filters, making them efficient for capturing spatial patterns and great for image data. However, CNNs struggle with capturing long-range dependencies and complex relationships in data, as their receptive field grows only incrementally with depth. Attention mechanisms, particularly in transformer-based models, address these limitations by enabling global context modeling. This global perspective has been useful in tasks requiring long-range dependencies, such as natural language processing and video understanding. By combining CNNs' efficient feature extraction with attention's capacity for global context, deep learning models can achieve state-of-the-art performance in many domains.

Our approach combines the unsupervised learning ability of a CNN-based auto-encoder and the token generation ability of a attention-leveraging neural network pre-trained on the Kinetics 600 dataset with the sequential data handling ability of an LSTM to create a more efficient method of identifying a variety of highlights from soccer games.

3. Related Work

Existing methodologies for highlight generation in soccer videos range from object detection and action recognition to advanced video summarization techniques.

For object detection and event detection, Faster R-CNN and YOLO have been applied in the context of soccer match analysis for detecting events like goals, fouls, and penalty kicks. Darapaneni et al. demonstrated that Faster R-CNN with ResNet50 as the backbone outperformed YOLO in terms of accuracy for event detection [1]. They reduced a 23-minute video to a 4:50-minute highlight reel, achieving 95.5% classification accuracy with Faster R-CNN [1]. Also, temporal pyramid pooling methods have been explored to enhance the granularity of temporal features in video frames. By segmenting videos into chunks and applying specialized pooling layers, researchers have improved the precision of action detection, ensuring better contextual understanding of soccer events [1].

Park et al. utilized approaches such as NetVLAD++ to employ temporally aware feature pooling for action spotting, focusing on detecting anchor events like goals and substitutions with high precision [2]. It has demonstrated improved mAP scores compared to baseline models such as CALF, making it useful for tasks requiring temporal context analysis.

TimeSformer, a transformer-based video understanding model, introduced divided attention mechanisms for spatiotemporal feature learning [3]. By adapting self-attention to operate across both spatial and temporal dimensions independently, TimeSformer achieved state-of-the-art performance in datasets like Kinetics-600. This model eliminates the need for convolutions, providing a scalable alternative for long-range dependencies in soccer highlight generation.

Our method builds upon these existing strategies by integrating the token generation capabilities of TimeSformer and VideoMAE models with LSTM networks for sequential dependency modeling. Unlike previous approaches, our approach explicitly makes use of recurrent architectures to handle the inherently sequential nature of the data. Additionally, unlike approaches reliant solely on predefined event classes or replay detection, our anomaly detection framework classifies mismatched predictions and observed actions as "surprises," contributing a novel dimension to highlight identification. This ensures a broader and more adaptive highlight generation process that incorporates unanticipated yet significant events. This integration of methodologies underscores the potential of combining advanced video understanding models with sequence learning to refine and enhance automated highlight generation.

4. Approach

Our system's approach consists of three main phases: (1) video classification on segments on a full soccer game, (2) training a LSTM on video classification tokens and sound intensity propagated through time, and (3) anomaly (highlight) detection using true and predicted tokens.

4.1: Video Classification

Our data consists of over one hundred full soccer matches at 25fps, each split into halves. For each half of a soccer match, we split the total 2700 seconds into 5-second segments with an overlapping window of 2.5 seconds. 5 second segments were chosen after testing with 1, 2, 5, and 10 seconds as 5 second segments eliminated significant noise present in the 1-2 second segments while still providing a significant number of classification tokens per video for training. A 2.5 second overlapping window was chosen to maintain continuity between video segments.

For the classification, we use two models: a TimeSformer model pretrained on the Kinetics-600 dataset, and a VideoMAE model also pretrained on the Kinetics dataset. The VideoMAE, or masked autoencoder, model is a video processor that performs self-supervised learning for video representations. The architecture focuses on learning compact representations by reconstructing heavily masked video inputs. VideoMAE applies extreme masking rates, leaving only approximately 10% of video patches visible during training. It uses a transformer-based encoder to learn the spatiotemporal relationships between visible patches and a lightweight decoder that reconstructs the full video. It uses a Mean-Squared Error (MSE) loss function on the difference between the pixels of the original and reconstructed videos. The TimeSformer model leverages the features learned by the VideoMAE model and their temporal structure via space-time attention to accurately classify each video segment. The TimeSformer model heavily leverages the self-attention mechanism and positional embeddings in both the spatial and temporal dimensions. It emphasizes long-range spatiotemporal modeling. As with many classification models, this model minimizes cross-entropy loss.

For each game half, the video classification works as follows. The game half is additionally split into two halves – one from frame 1 to 33750 and another from frame 33751 to frame 67500. This was done to solve a memory issue as the VideoMAE computations take up substantial space in memory that significantly slowed down prediction. For each half, we encode the 33750 frames with the VideoMAE model to learn the spatiotemporal relationships. After this, we split these 33750 pre-processed frames into segments of 5-second segments (125 frames) with a 2.5-second window of overlap. Each segment is then passed into the TimeSformer model which predicts a class token. This results in 1080 tokens per 45 minutes of play – the final output of this phase.

4.2: LSTM Training and Prediction

After extracting the tokens, a sequence length of 10 was chosen to train the LSTM, as this would allow the LSTM to predict the next token given the past $\sim$25 seconds. To take advantage of the multimodal nature of the videos and leverage its audio as a feature, we extract the sound intensity in decibels for every five-second interval, with each interval overlapping by 2.5 seconds, the same as how the tokens were processed. Since the absolute sound intensity varies by video due to factors such as different recording setups, we normalized loudness per video using z-score normalization. In the forward pass, we pass the token through an embedding layer with dimension 128 and concatenate the normalized decibel value at the end for an input dimension of 129 for each sample.

We added an additional parameter, which is the rate of change of token predictions. This signifies the speed of developments in the match and therefore is a proxy for the likelihood of a highlight; it is calculated by calculating the difference between token values in consecutive segments. A batch size of 32 was then used to build the training set of token sequences, audio data, and rates of change.

This data is then used to train the LSTM for next token prediction (prediction of the next action). The cross-entropy loss is used to calculate the loss of predictions over each batch, considering the sequence of tokens and audio data. Due to the heavy class-imbalance, label smoothing was used as a regularization technique to prevent the model from only predicting the majority class (kicking a soccer ball) every time. A weighting factor is then calculated based on the corresponding rates of change for each of the sequences in the batch, with higher rates of change resulting in a higher weighting factor. Each sequence's contribution to the loss is weighted by its weighting factor before stochastic gradient descent updates the LSTM weights via BPTT. A learning rate scheduler was also implemented to achieve faster convergence and improve stability.

We perform model validation in 10 batches over the validation set. Again, we compute the loss weighted by the rate of token changes in each validation sequence, and we store the tokens over which the model most frequently makes incorrect predictions. It is worth noting that due to our definition of a highlight (i.e. a notable surprise occurred), we do not necessarily desire the validation accuracy to be extremely high, otherwise a highlight would never be predicted.

4.3: Anomaly Detection

The last phase of our system involves anomaly detection and highlight generation, aiming to identify video segments that maximize a combined score based on **significance** and **surprise**, while incorporating contextual consistency and merging overlapping intervals.

We maximize the combined score over the selected segments S:

$$\max_{t \in S} \sum_{t \in S} \text{Highlight\_Score}_t, \quad \text{where} \quad \text{Highlight\_Score}_t = \frac{1}{w} \sum_{j=\max(1,t-\lfloor w/2 \rfloor)}^{\min(T,t+\lfloor w/2 \rfloor)} \big(w_1 \cdot \text{Norm\_Significance}_j + w_2 \cdot \text{Norm\_Surprise}_j\big).$$

Surprise Score (R_t) quantifies how unexpected the observed token y_t is, defined as:

$$\text{Surprise}_t = -\log P(y_t \mid \text{logits}_t), \quad P(c \mid \text{logits}_t) = \frac{e^{\text{logits}_t[c]}}{\sum_{k=1}^{600} e^{\text{logits}_t[k]}}.$$

Significance Score (S_t) measures the importance of a segment based on class-specific weights w_c, defined as:

$$\text{Significance}_t = \sum_{c=1}^{600} w_c \cdot P(c \mid \text{logits}_t), \quad w_c = \frac{1}{\text{class frequency in training data}}.$$

To make these scores comparable, min-max normalization is applied:

$$\text{Norm\_Significance}_t = \frac{S_t - \min(S)}{\max(S) - \min(S)}, \quad \text{Norm\_Surprise}_t = \frac{R_t - \min(R)}{\max(R) - \min(R)}.$$

Scores are smoothed to incorporate contextual information using a moving average:

$$\text{Highlight\_Score}_t = \frac{1}{w} \sum_{j=\max(1,t-\lfloor w/2 \rfloor)}^{\min(T,t+\lfloor w/2 \rfloor)} \big(w_1 \cdot \text{Norm\_Significance}_j + w_2 \cdot \text{Norm\_Surprise}_j\big),$$

where w is the smoothing window size.

A segment t is classified as a highlight if $\text{Highlight\_Score}_t > \tau$. τ, w_1, and w_2 are hyperparameters that can tailor the outputted highlight video to specific needs. Highlight intervals are converted to time intervals with a 10-second context before and after the segment. Overlapping intervals are merged to ensure no duplicate highlights in the final output.

5. Experimental Results

The images in figures 1 and 2 in the appendix outline the results of the video classification phase. As predicted, the most frequent tokens predicted include soccer related tokens such as shooting and passing as well as other general sports-related tokens like “pumping fist” and “headbutting”. Our least frequent predicted tokens illustrate the inherent noise that comes with this data. Actions such as “reading newspaper” and “skydiving” almost certainly do not occur in our games but may have been predicted due to irregular camera angles or shots. Following video classification, we trained an LSTM on the segmented token and audio sequences. The loss and accuracy over training are illustrated in figures 3 and 4 in the appendix. We noticed that training the model with sound intensity as a feature (concatenated at the end of the vector) resulted in a noticeable in accuracy, at nearly 5%. It also resulted in highlights that were able to take out more of the “boring” parts of the game and include most of the important events. We noticed that in general, the highlight generation was able to include all goals quite reliably.

Lastly, while sports highlights are inherently subjective, a novel approach to evaluation must be used. To do so, we randomly selected 3 games from our test dataset and used our system to generate a highlight reel for each. Then, our team polled 10 of our friends and asked them to rate how well each video captures the highlights on a scale of 1 to 7, with 7 being the highest. A scale of 1 to 7 was chosen for its proven optimality [4]. The results of this evaluation are shown below.

	Eval 1	Eval 2	Eval 3	Eval 4	Eval 5	Eval 6	Eval 7	Eval 8	Eval 9	Eval 10	Row Avg
Example Video 1	5	6	3	7	5	6	6	6	5	5	5.4
Example Video 2	4	3	5	3	5	5	4	7	6	3	4.5
Example Video 3	5	5	7	6	6	3	5	7	7	6	5.7
Column Avg	4.7	4.7	5	5.3	5.3	4.7	5	6.7	6	4.7	5.2

This method of evaluation is by no means perfect. There is inherent bias in the subjects selected as they are all part of a similar demographic (Penn students) and the sample size for both the number of games and the number of reviewers is quite small. However, given the scope of the project and time allotted, we believed this method would give us a small indication of the performance of our system.

6. DISCUSSION

In our limited qualitative testing, our model averaged a score of 5.2 out of 7 for its generated highlights across 3 videos and 10 evaluators. Given the many nuisances present in our dataset, including changing broadcast angles, crowd shots, and general inconsistencies in the stream, our team is pleased with these results. In our own testing, we observed that the model was very good at picking up general irregularities in the games – injuries, yellow and red cards, long stoppages of play, goals, substitutions, and more. "Regular" footage of play was never observed to be included in a single highlight reel, which is promising. However, being able to distinguish between these irregularities proved quite difficult. An occasional prolonged shot of a team's coach, a close-up of a player complaining to a ref, and an unimportant replay with many varying camera angles are a few examples among many of events that may not generally be considered "highlight worthy" that frequently made it into our output. While including the loudness factor into our LSTM certainly increased accuracy and resulted in slightly better highlight selection, it did not fully solve the aforementioned problems. Overall, we believe this LSTM-based approach for autonomous highlight detection has potential and could be promising given more time and resources.

6.1 Future Work

Increasing the quantity and quality of the data could help our model achieve greater results. For instance, obtaining additional modes of data (e.g. different broadcast options, commentator transcripts, etc.) could assist in piecing together a more complete picture of what constitutes a highlight. In addition, due to resource and timing constraints we relied heavily on a model pre-trained on the Kinetics 600 dataset to generate the sequential token data for our predictions. While our model generates reasonably high-quality highlights, we are limited by the fact that the Kinetics 600 tokens do not perfectly match up with the expected actions in a soccer game, which slightly alters our results. Better-labeled data that corresponds more explicitly to soccer could be used to test the resulting improvements in the prediction accuracy and quality of generated highlight clips.

In addition, we could experiment with different model architectures that could be better suited to soccer. V-JEPA, for instance, is a non-generative self-supervised model that learns by predicting missing or masked parts of a video in an abstract representation space. It's ability to generalize to a wider variety of videos and actions could be better suited for our use case. We could also experiment with attention-based architectures. While our LSTM yields quality results, the greater ability of attention-based architectures to handle very long sequences (such as soccer games) could contribute to even better accuracy and highlight quality.

Finally, more extensive qualitative testing could be beneficial for getting a better estimate of how well our model is generating highlights. A larger sample size of evaluators would give us a more comprehensive understanding of what the general population considers highlights. This could also allow us to tune our model to fit these generalized notions of a highlight.

References

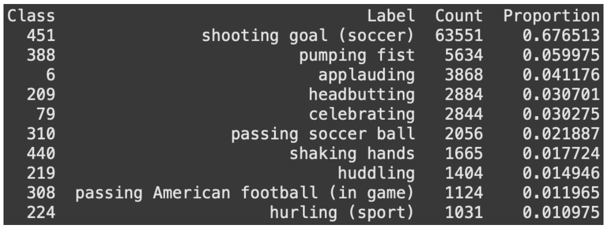
[1] N. Darapaneni, P. Kumar, N. Malhotra, V. Sundaramurthy, A. Thakur, S. Chauhan, K. C. Thangeda, and A. R. Paduri, ”Detecting key soccer match events to create highlights using computer vision,” arXiv preprint arXiv:2204.02573, Apr. 2022.

[2] J. Park, Y. Jwa, J. Kwak, J. Lim, and S. Kim, ”Automatic Highlight Generation of Soccer Videos,” in Proc. 14th International Conference on Information and Communication Technology Convergence (ICTC), 2023, pp. 1867–1871.

[3] G. Bertasius, H. Wang, and L. Torresani, ”Is Space-Time Attention All You Need for Video Understanding?” arXiv preprint arXiv:2102.05095, Jun. 2021.

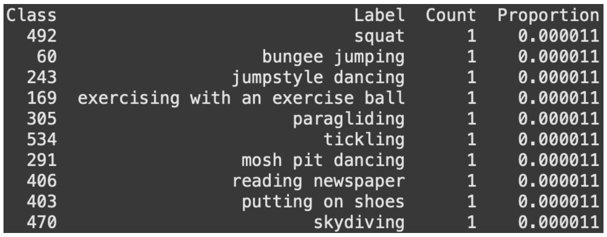
[4] J. R. Preston and J. E. Colman, ”Effect of the Number of Response Categories on the Reliability and Validity of Rating Scales”

APPENDIX



```
Class                          Label  Count  Proportion
  451          shooting goal (soccer)  63551    0.676513
  388                    pumping fist   5634    0.059975
    6                      applauding   3868    0.041176
  209                     headbutting   2884    0.030701
   79                     celebrating   2844    0.030275
  310               passing soccer ball   2056    0.021887
  440                    shaking hands   1665    0.017724
  219                        huddling   1404    0.014946
  308  passing American football (in game)  1124    0.011965
  224                  hurling (sport)   1031    0.010975
```

FIGURE 1. Most Predicted Classes



```
Class                          Label  Count  Proportion
  492                          squat      1    0.000011
   60                 bungee jumping      1    0.000011
  243              jumpstyle dancing      1    0.000011
  169  exercising with an exercise ball      1    0.000011
  305                    paragliding      1    0.000011
  534                       tickling      1    0.000011
  291                mosh pit dancing      1    0.000011
  406              reading newspaper      1    0.000011
  403               putting on shoes      1    0.000011
  470                      skydiving      1    0.000011
```

FIGURE 2. Least Predicted Classes (Count $>$ 0)

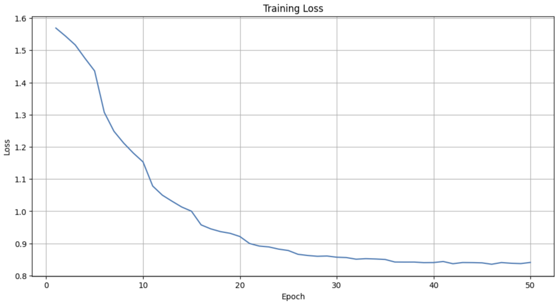


FIGURE 3. LSTM Training Loss vs. Epochs

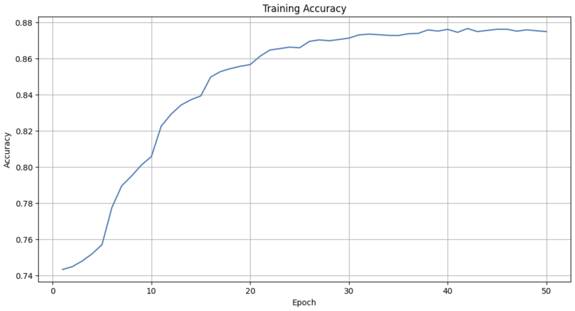


FIGURE 4. LSTM Training Accuracy vs. Epochs