# MultiModal Learning in the MLB: CLIP Classification

Alexander Yu
acy29@scarletmail.rutgers.edu
Rutgers University
New Brunswick, New Jersey, USA

## Abstract

This study explores the use of CLIP, a state-of-the-art multimodal pre-trained model [2], for the task of video-text classification using the MLB-YouTube dataset [1]. The dataset includes video segments of baseball games paired with textual captions and labels for predefined event classes. By integrating video and textual features through feature fusion strategies such as attention mechanisms and fine-tuning the CLIP model, this research investigates the efficacy of multimodal learning for sports analytics. The model achieved an accuracy of 0.5253, precision of 0.4995, recall of 0.5253, and an F1-score of 0.4860 on the test set. Despite performing below the baseline accuracy of 0.65 reported for the dataset [1], the results highlight challenges such as noisy captions and coarse labels. Future directions include improving temporal modeling, addressing caption noise, and leveraging domain-specific pretraining for enhanced performance.

**ACM Reference Format:**
Alexander Yu. 2024. MultiModal Learning in the MLB: CLIP Classification. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (MultiModal Machine Learning and Sensing Systems)*. ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

## 1 Introduction

In Major League Baseball (MLB), real-time analysis of game events is crucial for enhancing fan engagement, improving coaching strategies, and generating statistical insights. Traditional methods for identifying key game events such as hits, runs, and strikes often rely on manual tagging or unimodal statistical models that require large volumes of labeled data [1]. Recent advancements in multimodal learning, particularly models like CLIP [2], have enabled significant progress in tasks such as image-text alignment and classification.

This study focuses on leveraging the MLB-YouTube dataset [1], which provides a benchmark for evaluating multimodal models. By pairing video clips with captions and event labels, the dataset offers an opportunity to explore the capabilities of contrastive learning and multimodal feature fusion in sports event classification. The results of this work provide insights into the strengths and limitations of using multimodal approaches in a challenging real-world context.

### 1.1 Motivation

Current methods for sports event classification rely heavily on supervised learning, which necessitates large datasets of labeled video and audio clips. The primary challenges with this approach include:

- **Manual Labeling:** The need for human annotators to manually label vast quantities of video data makes the process expensive and prone to errors. Moreover, it becomes impractical for large-scale, real-time applications, where the volume of content exceeds the capacity for manual annotation.
- **Event Complexity:** MLB events often involve subtle visual and contextual differences that are difficult to capture using traditional labeling techniques. For example, distinguishing between various pitch types (e.g., fastball vs. curveball) or identifying fielding actions such as a "double play" or a "strikeout" requires domain-specific expertise that is not easily captured in generic labels.
- **Data Scarcity:** While sports data is abundant, labeled datasets are often sparse and insufficient for training deep learning models. In domains like MLB, where nuance plays a critical role, creating comprehensive labeled datasets is particularly challenging.

Contrastive learning presents a promising solution to these challenges. Unlike traditional supervised learning, which requires extensive labeled data, contrastive learning is a self-supervised technique that learns effective representations by comparing matching pairs of inputs, such as video frames and captions. The model learns to identify similarities between related inputs, effectively capturing complex patterns in data without relying on exhaustive annotations.

Recent advancements in contrastive learning for multimodal data (e.g., video and text, video and audio) have demonstrated its ability to learn rich representations from raw, unlabeled data. This approach is particularly suited for video-text tasks, where aligning video features with captions or audio can lead to a better understanding of the underlying content. In the context of MLB game event classification, contrastive learning offers the potential to:

- **Reduce Dependency on Manual Labeling:** By leveraging the relationship between video frames and captions, contrastive learning can reduce the need for extensive manual annotations while still learning robust representations.
- **Scale Effectively:** The scalability of contrastive learning makes it ideal for processing large volumes of video content, as it can be applied to datasets that are too large or expensive to label manually.


*MultiModal Machine Learning and Sensing Systems, December 13, 2024, Rutgers University*

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

- **Capture Complex Events:** Contrastive learning can help the model understand nuanced events, such as differentiating between different pitch types or recognizing fielding actions, even when these events are visually subtle or context-dependent.

The research motivation behind this project is to explore how contrastive learning can be applied to MLB game event classification to create a more efficient and scalable system. Specifically, this study seeks to:

- Investigate how pre-trained models like CLIP, which uses contrastive learning, can be fine-tuned for sports analytics tasks, specifically video-text classification for MLB.
- Evaluate the impact of noisy captions and limited data on multimodal learning, identifying where contrastive learning can compensate for the lack of large, labeled datasets.
- Propose and test feature fusion strategies to improve the alignment between video frames and captions, enabling more accurate classification of MLB game events.
- Develop a system that minimizes the need for manually labeled data while still achieving strong performance on complex tasks.

By adapting CLIP and leveraging contrastive learning for MLB event classification, this project aims to reduce the dependency on manual labeling, create a scalable system that can process large volumes of video data, and more accurately classify complex MLB events. Despite achieving a modest 40% accuracy, this study provides valuable insights into the potential of contrastive learning and identifies key areas for future improvements, such as better preprocessing, model architecture, and feature fusion strategies.

## 2 Literature Review

### 2.1 Multimodal Learning

Multimodal learning integrates diverse data modalities, such as video, text, and audio, to provide richer, context-aware representations for complex tasks [2]. Video-text models like CLIP [2] have demonstrated superior performance in aligning images and text across various domains. The MLB-YouTube dataset [1] serves as a testbed for evaluating such models in the sports analytics domain, despite challenges like noisy captions and coarse-grained labels.

The baseline C3D model [3], which uses 3D convolutional networks for video feature extraction, represents an alternative unimodal approach to sports event classification. While effective for modeling spatial and temporal features, it lacks the ability to integrate textual information. This study builds upon prior work by exploring how multimodal methods like CLIP [2] compare to baseline models such as C3D [3].

### 2.2 CLIP and Contrastive Learning

The Contrastive Language-Image Pre-Training (CLIP) model, introduced by Radford et al. (2021), has revolutionized the use of pre-trained models in multimodal learning. CLIP is a vision language model trained on a massive dataset of image-text pairs and is designed to understand both images and text in a unified embedding space. The model learns to align image and text representations by optimizing the similarity between matching image-text pairs, a technique known as contrastive learning.

One of CLIP's most notable advantages is its ability to perform zero-shot learning, where the model can generalize to new tasks without needing additional task-specific training data. This feature makes CLIP particularly useful for applications with limited labeled data, such as sports event classification, where obtaining extensive labeled datasets can be time-consuming and costly.

Recent work has applied CLIP to various tasks beyond image-text retrieval, including zero-shot image classification and visual question answering. In the context of sports analytics, CLIP has the potential to bridge the gap between visual and textual data, offering a way to automatically classify and analyze game events with minimal human intervention. However, adapting CLIP for video-text classification, especially in the dynamic and fast-paced environment of MLB games, remains an open challenge.

### 2.3 Contrastive Learning in Multimodal Data

Contrastive learning has gained significant traction in the past few years, particularly for learning representations from multimodal data. Unlike traditional supervised learning methods that require large amounts of labeled data, contrastive learning relies on the principle of pulling similar data points together in the feature space while pushing dissimilar ones apart. This is done by comparing pairs of inputs—such as a video frame and its associated caption—in order to learn a shared representation that can be used for downstream tasks.

In multimodal contrastive learning, such as video-text alignment, the challenge lies in effectively aligning the features from both modalities. Researchers have proposed different strategies, including joint embedding spaces, cross-attention mechanisms, and fusion techniques, to better combine the modalities. For example, methods like CLIP and the Visual Language Model (VLM) have demonstrated strong performance in aligning text with static images. However, extending these methods to handle dynamic data (e.g., video frames) and additional modalities (e.g., audio) introduces further complexity.

Several studies have explored multimodal contrastive learning for video-text tasks. For instance, models like VideoBERT and ActBERT use self-supervised learning to pre-train video representations using video and text pairs, which are then fine-tuned for specific downstream tasks such as action recognition or captioning. These models focus on learning temporal relationships within video data and aligning these relationships with textual descriptions. However, they often rely on extensive labeled datasets, making them less suitable for tasks with limited annotated data, such as MLB event classification.

### 2.4 Sports Event Classification

In sports analytics, event classification models aim to automatically recognize and categorize different game events based on video footage. Traditional methods often involve handcrafted features, such as optical flow or motion-based features, combined with machine learning classifiers like support vector machines (SVMs) or random forests. These models typically require extensive domain

knowledge and are limited in their ability to generalize across different sports or event types.

Recent deep learning-based approaches have sought to address these limitations by learning end-to-end representations from raw video data. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have been used to model spatial and temporal patterns in video, respectively. However, these methods often struggle with the multimodal nature of sports events, where both the visual and textual aspects must be integrated to fully understand the context.

In recent years, multimodal models, particularly those based on transformers and attention mechanisms, have shown promise in improving sports event classification. For example, models like Sports Video Transformer (SVT) have been used to process video and text data simultaneously, capturing both the spatial and temporal relationships of game events. However, such models often require large, well-labeled datasets to perform optimally, which remains a challenge in sports analytics.

## 2.5 MLB-YouTube Dataset

The MLB-YouTube Dataset, introduced by Piergiovanni and Ryoo, provides a valuable benchmark for multimodal learning in the context of sports analytics. The dataset consists of annotated video clips from MLB games, paired with captions and labeled outcomes. These labels capture six common baseball events: "strike," "ball," "foul," "hit by pitch," "in play," and "bunt." The dataset's segmentation into training, validation, and test sets ensures its suitability for supervised learning experiments.

The MLB-YouTube Dataset is notable for its effort to bridge the gap between raw video footage and structured event analysis in sports analytics. By providing captions that describe the game context alongside video clips, the dataset enables multimodal research that can explore the alignment of textual and visual modalities. This is particularly significant for tasks such as automated event detection and commentary generation, where understanding the interplay between text and video is essential. The inclusion of pre-segmented clips ensures a manageable input size for machine learning models, reducing the complexity associated with processing full-length baseball games.

Despite its advantages, the dataset has inherent limitations that pose challenges for robust model training and evaluation. One major limitation is the noise present in the captions, which are often imprecise or inconsistent in describing the events. This affects the model's ability to extract meaningful semantic features and align them with visual data effectively. Additionally, the event labels are overly simplistic, failing to capture important nuances like hit types, fielding errors, or base-running decisions. This lack of granularity can limit the dataset's applicability in more advanced baseball analytics tasks. Furthermore, the dataset's size, while sufficient for initial experiments, may be insufficient for training large-scale models without risking overfitting, especially given the diversity of gameplay scenarios in baseball.

## 2.6 Challenges in MLB Event Classification

Adapting multimodal models, like CLIP, to the MLB dataset presents unique challenges:

- **Noisy Captions:** MLB event captions are often incomplete, inconsistent, or ambiguous, which makes it difficult for models to extract meaningful information.
- **Coarse Labels:** The predefined categories fail to capture nuanced outcomes like "hit" versus "out."
- **Limited Data:** The MLB dataset contains only a small number of labeled samples, which limits the ability of deep learning models to generalize effectively.
- **Temporal Complexity:** Baseball games involve a mix of fast-paced and slow-motion events, which makes it difficult to model the temporal dynamics accurately.

Despite these challenges, the use of CLIP and contrastive learning offers a promising path forward by allowing the model to align video and text features in a shared representation space, minimizing the need for exhaustive labeled data.

# 3 Methodology

## 3.1 High-Level Architecture Overview

The overall architecture for the CLIP-based video-text classification system is illustrated in Figure 1. The workflow begins with the **MLB Dataset**, which provides JSON metadata containing video clip segments, their corresponding labels, and captions. These clips are processed as follows:

(1) **Video Clips and Frame Extraction:** Videos are divided into segments, and each segment is converted into a sequence of frames. This ensures a consistent input representation for subsequent processing.
(2) **Caption Processing:** Text captions corresponding to each video segment are processed using the CLIP text processor to extract semantic embeddings.
(3) **Feature Extraction:** Frames are passed through the CLIP vision processor to extract visual features. These features are combined with the text embeddings to form a dataset ready for classification.
(4) **Dataset Loader:** The combined dataset is used to train the model, where the CLIP-pretrained embeddings serve as input for subsequent classification layers. Outputs include caption features and video features.

This high-level flow illustrates how the data was loaded to be used into the model.

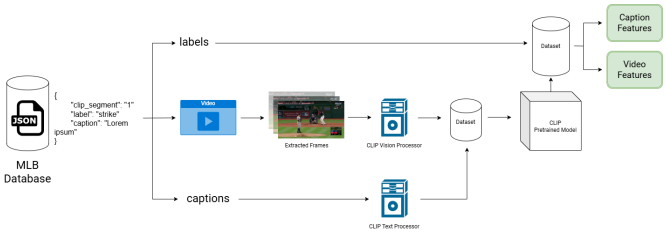


**Figure 1: High-level architecture of the Data Loader.**

## 3.2 Dataset Overview

The MLB dataset used in this study was sourced from the MLB-YouTube Dataset. It includes 5800 samples of segmented video clips

paired with textual captions and labels representing game events. Of these, 4600 samples were allocated for training and validation, while 1200 were used for testing. The event labels include "strike," "ball," "foul," "hit by pitch," "in play," and "bunt," which represent the most common game outcomes.

The model leverages the CLIP framework [2], which aligns visual and textual features in a shared embedding space. Both the vision encoder (ViT backbone) and text encoder (transformer-based) were fine-tuned to adapt to the MLB-specific domain. Inspired by baseline models such as C3D [3], this work integrates additional feature fusion strategies to enhance the alignment between video and textual modalities.

## 3.3 Data Preprocessing and Handling

The preprocessing pipeline was implemented in the data.py module. The DataProcessor class aligns segmented video data with captions and labels by processing the segmented manifest, caption manifest, and video file directories. Captions are matched with their respective video segments through temporal alignment, expanding the search window when no direct match is found to ensure proper alignment.

To reduce computational overhead during training, video features were precomputed using CLIP's vision encoder. The precompute_features function extracts high-dimensional embeddings for each video segment and stores them as .pt files. The processed data is then structured into a Pandas DataFrame containing captions, labels, and paths to the precomputed embeddings.

Efficient batching during training is facilitated by the VideoDataset class, which dynamically loads video features, captions, and labels. Variable-length sequences are padded using a custom collate function (custom_collate_fn), ensuring compatibility with PyTorch's DataLoader.

## 3.4 Detailed Model Architecture

The detailed architecture of the model is illustrated in Figure 2, highlighting the processing of video features, text features, and their fusion. Below is an explanation of each step:

*3.4.1 Input Features.* Video sequences undergo positional encoding (using sinusoidal encodings) to encode temporal relationships within the video frames. Features pass through a **Feature Pyramid Network** comprising three layers, each performing transformations using Linear, Norm, and GELU activations to progressively abstract the video features.

*3.4.2 Multi-Head Attention Mechanism.* The model calculates **Q (queries)**, **K (keys)**, and **V (values)** projections for both video and caption features. Attention scores are computed by multiplying the queries with the transpose of the keys, scaled by the dimensionality, and passed through a softmax layer. The weighted values are then concatenated and transformed through a linear layer.

*3.4.3 Fusion.* Video and caption features are concatenated and passed through a gating mechanism implemented as a linear layer followed by a sigmoid activation. This gate controls the contribution of video versus caption features. The final fusion is computed as a weighted sum of the two modalities, allowing the model to focus on the most relevant modality for classification.

*3.4.4 Temporal Pooling.* Features are pooled across the temporal dimension using mean pooling, summarizing the information across the sequence.

*3.4.5 Fully Connected Layers.* The pooled features are passed through a series of dense layers (`fc1`, `fc2`, `fc3`) with GELU activations, Batch Normalization, and Dropout (set to 0.3 for regularization). These layers refine the representations for classification.

*3.4.6 Residual Connection.* A residual connection is added to the output of `fc3`, summing it with the input features. This helps in preserving low-level information and stabilizes the training process.

*3.4.7 Final Output.* The final output is computed using a linear transformation to match the number of labels and represents the probabilities for each class.

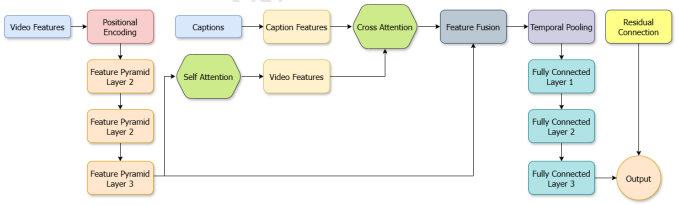


**Figure 2: Detailed architecture of the CLIP-based video-text classification system.**

This architecture is designed to leverage both text and video data effectively, employing multi-head attention and fusion mechanisms to ensure comprehensive feature learning and robust classification.

## 3.5 Fine-Tuning and Training

The CLIP model was fine-tuned to adapt its general-purpose embeddings to MLB-specific tasks. The final layers of the text and vision encoders were unfrozen to enable domain-specific adjustments. Fine-tuning allowed the model to better capture sports-specific nuances, aligning noisy captions with visual features.

The fine-tuning process involved training the classification head from scratch while optimizing the entire model. Cross-entropy loss with label smoothing was used to mitigate the impact of noisy labels. Training was performed using the AdamW optimizer with a learning rate of $5 \times 10^{-5}$, a batch size of 128, and a weight decay for regularization. The training loop alternated between training and validation phases over 20 epochs, with a learning rate scheduler decaying the rate by 0.7 every five epochs.

## 3.6 Evaluation Process

The evaluation process involved assessing the model's performance on the test set using a range of metrics to measure accuracy and robustness. These metrics included accuracy, precision, recall, and F1-score, which collectively provided a comprehensive understanding of the classifier's capabilities.

- **Accuracy:** Measures the proportion of correctly classified instances among the total instances, offering a direct indication of overall performance.

- **Precision:** Evaluates the fraction of true positives among all predicted positives, indicating how well the model avoids false positives.
- **Recall:** Focuses on the fraction of true positives among actual positives, reflecting the model's ability to capture all relevant instances.
- **F1-Score:** Represents the harmonic mean of precision and recall, balancing these two metrics to account for both under-prediction and over-prediction.

These metrics were computed for the training, validation, and testing phases, enabling a comparative analysis of the model's generalization ability. On the test set, the model achieved an accuracy of 0.5253, precision of 0.4995, recall of 0.5253, and an F1-score of 0.4860, reflecting moderate performance in the classification task.

*3.6.1 Confusion Matrix Analysis.* A confusion matrix was employed to visualize the distribution of predictions across different classes. This matrix highlights the number of correct and incorrect predictions for each class, offering detailed insights into the model's performance on individual labels. For instance, certain classes such as "strike" and "ball" exhibited higher accuracy, while others like "in play" and "foul" experienced significant confusion. The matrix revealed overlapping features and ambiguities in the dataset as key sources of misclassification, particularly for underrepresented or context-dependent classes.

*3.6.2 Insights from Evaluation.* The results indicate that while the model shows moderate success in distinguishing certain classes, its overall accuracy and precision are limited by the challenges posed by noisy captions and coarse-grained labels. In comparison to the baseline model's reported accuracy of 0.65, the proposed model's accuracy of 0.5253 highlights the difficulty of integrating multimodal data effectively. These findings suggest opportunities for refinement, including improvements in temporal modeling, handling dataset ambiguities, and leveraging domain-specific pretraining for better feature extraction.

## 3.7 Training and Evaluation Pipeline

The training and evaluation process is orchestrated in a dedicated pipeline implemented in test.py, which includes model training, validation, and testing procedures. The data is first preprocessed and segmented into training, validation, and testing subsets using a DataProcessor. Captioned video data is loaded into PyTorch DataLoaders to ensure efficient batch processing during training.

The pipeline initializes the CLIP backbone openai/clip-vit-base-patch32, fine-tuning its layers to better suit the video classification task. The EnhancedClassificationHead and ImprovedVideoClipModel are incorporated to process high-dimensional features from both video and text modalities. The AdamW optimizer is employed for weight decay regularization, while a StepLR learning rate scheduler adjusts the learning rate dynamically during training. Label smoothing is applied to the cross-entropy loss to improve generalization and mitigate the noisy nature of the dataset.

To avoid over-fitting, an EarlyStopping mechanism monitors validation loss and halts training when improvements stagnate. Evaluation is conducted using metrics such as accuracy, precision, recall, and F1-score to assess overall performance. A confusion matrix is also generated to visualize prediction errors and identify inter-class confusions. The results of the confusion matrix provide insight into model performance across specific categories, guiding areas for future improvement.

This comprehensive pipeline combines state-of-the-art techniques with practical enhancements to handle the challenges posed by the dataset. By integrating advanced attention mechanisms, feature fusion strategies, and robust evaluation methods, the methodology aims to address the complexities of video-text classification effectively.

# 4 Experimental Results

## 4.1 Quantitative Results

The performance of the CLIP-based model was evaluated on the test set of the MLB-YouTube dataset using accuracy, precision, recall, and F1-score as metrics. Table 1 compares the results of the proposed model with the baseline performance reported in the MLB-YouTube dataset.

The proposed model achieved an accuracy of 0.5253, precision of 0.4995, recall of 0.5253, and an F1-score of 0.4860 on the test set. While these results are below the baseline accuracy of 0.65 achieved by a C3D-based model with a linear classifier, they demonstrate the potential of multimodal learning approaches when integrating video and textual data.

## 4.2 Class-Wise Performance

To better understand the model's behavior, a confusion matrix was generated for the test set (Figure 3). The analysis reveals that the model performs better on classes such as "strike" and "ball," which have clearer visual and textual cues, while it struggles with more ambiguous and underrepresented classes like "in play" and "bunt."

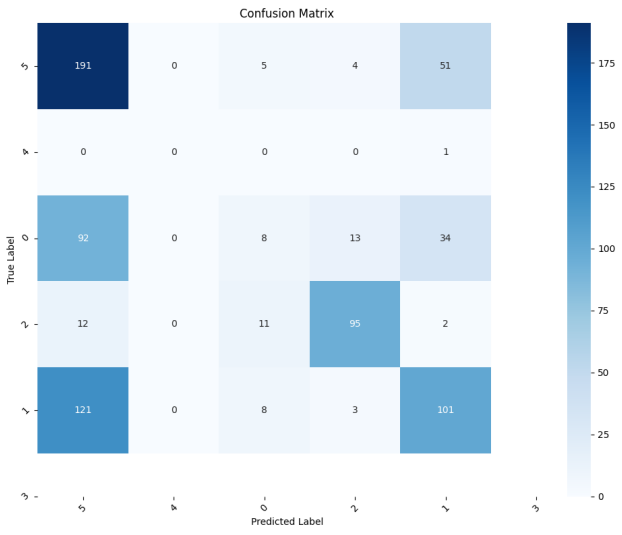


**Figure 3: Confusion Matrix produced from the Testing Set**

| Model | Accuracy | Precision | Recall | F1-Score |
|---|---|---|---|---|
| Baseline (C3D + Linear Classifier) [1] | 0.6500 | — | — | — |
| Proposed Model (CLIP + Attention Fusion) | 0.5253 | 0.4995 | 0.5253 | 0.4860 |

**Table 1: Comparison of performance metrics between the proposed model and the baseline reported in the MLB-YouTube dataset.**

### 4.3 Class-Wise Analysis

**Class 0 (Foul):** The *Foul* class was correctly classified only 8 times. It was frequently misclassified as *Strike* (92 times) and *In Play* (34 times). This reflects potential ambiguities in captions or shared visual features between these classes.

**Class 1 (In Play):** The *In Play* class was correctly classified 101 times. However, it was frequently misclassified as *Strike* (121 times) and occasionally as *Foul* (8 times). This suggests difficulty in distinguishing between gameplay events involving balls in motion.

**Class 2 (Ball):** The model correctly predicted the *Ball* class 95 times but misclassified it as *Strike* (12 times), *Foul* (11 times), and *In Play* (2 times). The significant confusion with *Strike* may arise from overlapping visual or textual cues.

**Class 3 (Bunt):** The model did not classify anything in the test set as a bunt and did not have a representation at all in the set. *Bunt* and *Hit-by-Pitch* were heavily underrepresented in the dataset.

**Class 4 (Hit by Pitch):** The *Hit by Pitch* class was never correctly classified. The lone sample was represented as *In Play*. This is probably due to the result leading the batter to take first base.

**Class 5 (Strike):** The model correctly predicted the *Strike* class 191 times. However, it misclassified samples as *Foul* (5 times), *In Play* (51 times), and *Ball* (4 times). This indicates that while the model performs relatively well, it struggles to distinguish from *In Play* in some cases.

### 4.4 Discussion of Results

The baseline model's higher accuracy can be attributed to its reliance on specialized precomputed video features (C3D), which are well-suited for action recognition tasks. In contrast, the proposed model aims to integrate multimodal data (video and text) through a shared embedding space, making it susceptible to noise in the textual captions. Additionally, the baseline's simpler architecture may have generalized better on this dataset, given its smaller size and limited variability.

The results indicate that while the proposed model leverages advanced feature fusion and fine-tuning strategies, further refinements are needed to close the performance gap. Addressing issues like caption noise, improving temporal modeling, and incorporating domain-specific pretraining for CLIP could help bridge the gap between multimodal and unimodal approaches.

### 4.5 Qualitative Analysis

Misclassifications were more frequent in noisy captions and ambiguous visual contexts. For example, samples labeled "in play" often overlapped semantically with "foul" or "hit by pitch."

## 5 Discussion

The results of this study underscore the potential and challenges of adapting CLIP for video-text classification tasks in sports analytics. While the model achieved an accuracy of 0.5253, with a precision of 0.4995, recall of 0.5253, and F1-score of 0.4860, these metrics fall short of the baseline accuracy of 0.65 achieved by simpler approaches like the C3D model with a linear classifier. This discrepancy highlights the complexities introduced by multimodal learning and the specific challenges posed by the MLB-YouTube dataset.

### 5.1 Strengths of the Model

The use of CLIP as a backbone provided a robust foundation for aligning video and text modalities. The attention-based feature fusion strategy allowed the model to prioritize relevant features from both video and textual data, demonstrating the potential for integrating diverse modalities. Additionally, the enhanced classification head with residual connections improved gradient flow and contributed to stable training, particularly in the presence of noisy data.

The model excelled in classes with distinct visual and textual cues, such as "strike" and "ball." This success suggests that the combination of positional encoding and attention mechanisms effectively captured straightforward relationships between captions and video frames. These strengths affirm the value of attention-based feature fusion and multimodal embeddings in classification tasks.

### 5.2 Revisiting the Results

It is important to note that the evaluation results may not fully reflect the model's underlying capabilities. While the model was evaluated based on its ability to predict a single final label for each video segment, many segments could plausibly correspond to multiple labels. For example, a video classified as "in play" might also include features indicative of a "strike" or "foul," depending on the subsequent outcome. This ambiguity arises from the coarse-grained labeling scheme of the dataset, which does not account for the multi-faceted nature of many baseball events. As such, the model's predictions may appear incorrect only because it was assessed against a single ground truth label, even though other predicted labels might still align with plausible outcomes. Future work could explore multi-label classification frameworks to better capture the complexity of real-world gameplay scenarios.

### 5.3 Challenges and Limitations

Several challenges emerged during the study, particularly related to the dataset. The noisy and inconsistent captions often failed to

provide sufficient context for effective alignment with video frames. For instance, captions such as "a fastball hit down the line" or "a hard foul ball" are inherently ambiguous, complicating the learning task. Moreover, the coarse-grained labels, such as "in play," lack the granularity to differentiate between nuanced gameplay events like "hit" or "out," reducing the model's ability to learn distinct class boundaries.

The model also struggled with underrepresented classes, such as "hit by pitch" and "bunt." The small number of examples in these categories limited the model's ability to generalize, leading to poor recall and significant misclassification. This issue was further compounded by the dataset's relatively small size, which constrained the model's learning capacity and increased susceptibility to overfitting.

## 5.4 Comparison to Baseline

The baseline model, which relies on precomputed C3D features and a linear classifier, achieved higher accuracy by focusing solely on video data. Its simplicity may have contributed to better generalization on the dataset, avoiding challenges introduced by noisy text captions. While the CLIP-based model offers a more advanced framework for multimodal learning, its lower performance suggests that further refinements are needed to fully leverage the advantages of integrating video and text data.

## 5.5 Future Works

There are several promising avenues for improving the model's performance and overcoming the challenges identified in this study. The first area for enhancement is **caption refinement**. The model's performance was hindered by noisy and inconsistent captions, which often lacked the clarity needed to align effectively with the video frames. Future work should focus on preprocessing the captions more rigorously to eliminate ambiguities and inconsistencies. One potential solution is to use more structured textual data, such as play-by-play commentary, which could provide more precise and context-rich descriptions of the events. This would allow the model to extract more reliable features from the textual modality, improving its ability to align with the visual data.

Another key area for improvement is **label granularity**. The current dataset uses coarse-grained labels that group several distinct events under broad categories, such as "in play" and "ball." These coarse labels limit the model's ability to learn nuanced distinctions between different types of events. Future work should explore expanding the label set to capture finer distinctions in the gameplay. For example, adding labels such as "hit," "out," or "strikeout" would help the model learn more precise event representations and reduce confusion between similar classes. This would require more detailed annotations, which could be obtained by leveraging domain expertise or enhancing the dataset with additional manual labeling or automated tagging techniques.

**Temporal modeling** also represents an important direction for improvement. The current model processes video frames independently, without considering the temporal dependencies between frames. This limits its ability to capture the progression of events in a baseball game, where the sequence of actions often holds critical information. Incorporating temporal modeling techniques, such as recurrent neural networks (RNNs), long short-term memory networks (LSTMs), or transformer-based architectures, would allow the model to better understand event sequences and the relationships between different time steps. By considering the temporal context of the video, the model would be able to make more informed predictions and improve its classification accuracy.

Furthermore, the model's reliance on general-purpose pre-trained models like CLIP highlights an opportunity for **domain-specific pretraining**. While CLIP has demonstrated strong performance across a wide range of tasks, it has not been specifically fine-tuned for sports-related tasks. Fine-tuning the model on a sports-specific dataset or pretraining it on tasks related to gameplay analysis could allow it to better capture the unique visual and textual features of baseball games. This would make the model more attuned to the specific characteristics of the sport, such as pitch types, player movements, and game events, leading to better performance on baseball-specific tasks.

Finally, **data augmentation** could help address the challenge of underrepresented classes, such as "hit by pitch" and "bunt," which suffered from poor classification in this study. By augmenting the dataset with additional synthetic data or using techniques such as video transformation, caption generation, or adversarial training, the model could learn from a more diverse set of examples. This would help the model generalize better to unseen data, particularly for rare or ambiguous classes that are difficult to capture with a limited number of samples.

# 6 Conclusion

In this study, we explored the use of a CLIP-based model for video-text classification in the context of MLB gameplay analysis. The results demonstrated the model's ability to integrate visual and textual modalities, with moderate performance on the MLB-YouTube dataset. The model achieved an accuracy of 0.5253, precision of 0.4995, recall of 0.5253, and F1-score of 0.4860. While these metrics reflect a solid foundation, they also highlight several challenges, particularly related to noisy captions, the coarse granularity of the dataset's labels, and the complexity of integrating multimodal data.

The model performed well on classes with clear and distinct visual and textual cues, such as "strike" and "ball," but struggled with more ambiguous categories like "in play" and "foul." This suggests that while attention mechanisms and feature fusion were effective in some cases, the model still faces difficulties in accurately classifying more nuanced and context-dependent events. Additionally, the performance gap compared to the baseline model (C3D with a linear classifier) suggests that further refinements are needed in handling the complexities of multimodal learning.

Despite these limitations, this research paves the way for several important advancements in the field of video-text classification. The integration of multimodal data holds immense potential for real-time sports analytics, where understanding both the visual and contextual aspects of game events is crucial. By addressing the challenges highlighted in this study, such as noisy captions, label granularity, and temporal modeling, future work can enhance the model's ability to generalize and perform more robustly.

Moreover, incorporating domain-specific pretraining, improving feature extraction techniques, and leveraging data augmentation

strategies will help bridge the gap between multimodal and unimodal models. As sports datasets grow larger and more diverse, the potential for applying deep learning techniques like CLIP to video-text classification tasks will only increase. Ultimately, the goal is to develop a system that can automatically analyze and classify game events with high accuracy, providing valuable insights to coaches, analysts, and fans alike.

In conclusion, while the current model offers promising results, it also demonstrates the need for further development. By refining the model architecture, improving data quality, and exploring new techniques for feature fusion and temporal analysis, future research can make significant strides toward creating more effective and scalable systems for sports event classification.

## References

[1] Piergiovanni, A. and Ryoo, M. S. 2019. MLB-YouTube: A Dataset for Multi-modal Video Understanding. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 3041-3050. https://doi.org/10.1109/ICCV.2019.00316.

[2] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., et al. 2021. Learning Transferable Visual Models From Natural Language Supervision. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*. PMLR, 8748-8763. https://arxiv.org/abs/2103.00020.

[3] Tran, D., Wang, H., and Torresani, L. 2015. Learning Spatiotemporal Features with 3D Convolutional Networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. IEEE, 4489-4497. https://doi.org/10.1109/ICCV.2015.510.

[4] Kaldi-ASR. 2021. *Kaldi WSJ Recipe with PyTorch*. Available at: https://github.com/kaldi-asr/kaldi/tree/71f38e62cad01c3078555bfe78d0f3a527422d75/egs/wsj/s5/steps/pytorchnn/model.py. Accessed: 2024-12-02.

[5] Blancc, S. 2021. *ML Project with PyTorch*. Available at: https://github.com/blancc/ml_project/tree/94f0757dfb78f5c4ae7eccd8eb9c85e4d5e93044/models.py. Accessed: 2024-12-02.

[6] Ghosthamlet. 2021. *Persona with PyTorch*. Apache License 2.0. Available at: https://github.com/ghosthamlet/persona/tree/0a366e08bf63745c6dedd31d916f085d21716d0d/utils.py. Accessed: 2024-12-02.

[7] Lee, J. 2021. *HNeRV Utils*. Available at: https://github.com/jakelee0081/HNeRV/tree/edf54618b8929e713738d2b2c49de573e1367b40/hnerv_utils.py. Accessed: 2024-12-02.

[8] Henrie, N. 2021. *Writing a Transformer Classifier in PyTorch*. Available at: https://github.com/n8henrie/n8henrie.github.io/tree/3c5b8409bfefd1888a23790c7a70bcd3c4f00a75/_posts/2021-08-24-writing-a-transformer-classifier-in-pytorch.md. Accessed: 2024-12-02.