DL4DS AI-Powered Cricket Commentary: Event Detection, GPT-4o Narration, and gTTS Synthesis

Bhuvan Shivalingaiah Gowda, Samritha Aadhi Ravikumar, Sumanth Hosdurg Kamath

May, 2025

Abstract

This project develops independent modules toward automating cricket commentary generation. In this work, we focus on generating cricket commentary using OpenAI's GPT-4o vision model by passing sampled video frames to produce event-specific textual narrations. The generated commentary is further synthesized into speech using the gTTS model, setting the groundwork for a future end-to-end automated cricket broadcasting system.

Introduction

Generating real-time cricket commentary from video highlights presents an exciting opportunity to combine computer vision and natural language generation technologies. Traditional cricket commentary requires human expertise to interpret game events and narrate them in an engaging manner. Automating this task can enhance sports broadcasting, enable real-time multilingual commentary, and create new forms of fan engagement.

In this project, we focus specifically on generating cricket commentary using OpenAI's GPT-4o Vision model. Sampled frames from cricket highlight videos are passed to the model to generate context-aware, concise commentary in the style of a professional commentator. The resulting textual commentary is then converted into speech using the gTTS model, creating a modular framework for automated cricket voiceovers. Our work forms an essential part of a broader effort to develop an end-to-end pipeline for intelligent sports narration.

The code, sample notebooks, and instructions for reproducing the commentary generation experiments can be found in the project GitHub repository: https://github.com/sumanth-1810/DS-542-AI-Cricket-Commentary-Generator

Related Work

Undergraduate Project Work:

Previous efforts in automated cricket commentary systems have largely focused on extracting structured information from video frames. In a prior project titled Cricket Commentary Subtitle Generator, we developed a pipeline that extracted frames from cricket videos and used EasyOCR to detect the ball number displayed on the scoreboard within each frame. This ball number was then matched to a pre-existing commentary text dataset, and the corresponding commentary was superimposed onto the video frames as subtitles. This approach allowed for synchronization of commentary with game events, enhancing the viewing experience for cricket enthusiasts and content creators.

While the Cricket Commentary Subtitle Generator successfully automated the generation of video subtitles, it relied on the availability of pre-existing commentary texts and accurate scoreboard readings. It did not involve interpreting gameplay or generating original narration based on raw visuals. Although the code and results were made reproducible and open for extension, the system was limited in its ability to dynamically understand or describe unseen game events.

In contrast, the current project leverages the GPT-4o Vision model to directly analyze sequences of frames and generate original, human-like cricket commentary without relying on external textual sources. This represents a significant evolution toward true vision-language-based cricket narration, moving beyond static subtitle generation to dynamic commentary generation.

Bundesliga–AWS Commentary System

In a recent advancement toward automated sports commentary, the Bundesliga partnered with Amazon Web Services (AWS) and Sportec Solutions AG to build a real-time generative AI commentary system [6]. The pipeline leverages structured match data (such as passes, goals, and fouls) ingested from live feeds and processed using AWS Lambda for event recognition and orchestration. Amazon Bedrock is used to interface with foundation models that generate commentary in multiple tones—ranging from "Sports Journalist" to "Casual" and "Gen Z"—allowing for style variation and audience personalization. The solution integrates with Amazon Translate and Amazon Polly for multilingual and text-to-speech support, respectively, enabling scalable commentary in over 20 languages. By combining low-latency event processing with large language model outputs, this approach showcases how multimodal AI pipelines can enhance viewer engagement and accessibility in sports broadcasting. The use of cloud-native tools also ensures real-time scalability and fault-tolerant deployment, making it suitable for high-traffic match days.

The Masters–IBM AI Commentary

IBM collaborated with Augusta National Golf Club to launch an AI-powered commentary system for The Masters golf tournament [7]. This system utilized generative AI to automatically generate spoken commentary for over 20,000 video clips during the tournament. The AI was trained on a large corpus of golf-specific language, enabling it to generate human-like narration that mimics traditional golf commentary. Leveraging IBM Watson technologies—including natural language processing and text-to-speech—the system could detect the context of each golf shot (e.g., a chip-in, birdie, or bunker save) and narrate it accordingly. Importantly, it could scale across all players and shots, ensuring comprehensive coverage regardless of broadcast priorities. This work exemplifies how domain-specific language modeling combined with event tagging can yield scalable commentary generation across thousands of sports events, making it an important precedent for cricket and other sports applications.

Murata – Generative AI in Sports Commentary

Murata's article highlights the growing impact of generative AI in transforming the sports viewing experience by automating live commentary generation [8]. It outlines how vision-language models can analyze video streams in real time to identify events such as goals, fouls, or milestones, and then generate corresponding textual or audio commentary. The article discusses applications ranging from enhancing fan engagement during live broadcasts to supporting AI-based coaching assistants and "AI caddies" in sports like golf. These systems rely on a combination of computer vision for real-time event recognition and natural language generation models trained on domain-specific data. This approach mirrors our project's vision of applying multimodal AI to automate commentary generation in cricket, demonstrating the broader industry trend toward scalable, AI-driven sports media.

Methodology

Part 1 - Event Recognition Model

To detect and isolate individual cricket events from highlight videos, we implemented a motion-based event recognition pipeline using OpenCV and MoviePy. The methodology follows a lightweight yet effective visual segmentation strategy:

- **Frame Difference Scoring**: Consecutive video frames are converted to grayscale, and their absolute pixel-wise differences are computed to quantify visual motion.
- **Top-K Scene Selection**: Instead of using a static threshold (which led to over-segmentation), we compute frame-wise difference scores across the video and extract

the top K frames with the highest visual change, typically corresponding to ball deliveries or transitions.

- **Minimum Gap Filtering**: To prevent closely clustered frames from being misclassified as multiple events, a 3-second temporal buffer is applied to enforce spacing between detections.

This process allows the commentary generation model to receive clean, self-contained visual inputs for each event, enabling better context retention and narration quality.

Part 2 - GPT Vision Model

Frame Extraction and Preprocessing

[1]To prepare inputs for OpenAI's GPT-4o vision model, each cricket highlight video is decoded using OpenCV and converted into individual RGB frames. Since the GPT-4o Vision API expects image inputs in base64-encoded format, each frame is compressed to JPEG and base64-encoded using Python's `cv2.imencode` and `base64` modules. To reduce computational overhead and ensure that token limits (input image tokens + text prompt tokens) are not exceeded, a sampling strategy is employed. Every Nth frame is selected (where $N \in \{15, 20, 30, \ldots, 100\}$), depending on the visual density and temporal duration of the event. This ensures that sufficient visual cues are retained while optimizing API usage.

Prompt Engineering

The GPT-4o vision model relies heavily on precise prompting to understand the intended task.[2] We design our prompts to minimize ambiguity and explicitly instruct the model to generate cricket-specific commentary. Two categories of prompts are used:

- **Single Event Prompts**: For isolated clips containing one delivery, the prompt instructs the model to provide a single concise commentary line, describing only the delivery type, shot played, and outcome (e.g., four, six, wicket).
- **Multiple Events Prompts**: For continuous clips with several deliveries, the prompt guides the model to identify and number each commentary line based on the visual sequence, such as:

  > "This video contains multiple cricket deliveries. Please provide a separate short commentary for each delivery, numbered 1., 2., 3., etc."

[3]Prompts are carefully designed to avoid model hallucination by discouraging mention of player names, crowd details, or specific camera views. This constraint ensures that the generated output focuses on gameplay dynamics inferred solely from the input frames.

Model Inference

[4]Once the base64-encoded frames and prompt are bundled into the input payload, they are passed to either the `gpt-4o` or `gpt-4o-mini` model using OpenAI's Chat API. The inference call sets a `max_tokens` constraint (typically between 100–200) to control output length and reduce latency. The model processes the prompt-image context using its internal vision-language transformer layers, aligning spatial cues across frames with linguistic features to generate coherent cricket commentary.

The output, returned as a structured JSON response, contains fluent, match-relevant commentary—often mimicking the tone and brevity of a human commentator.

Ablation Study

To identify the minimal number of visual inputs required per event, we conducted an ablation study by varying the frame sampling interval. For a given single-event clip (typically 3–5 seconds long), we tested frame sampling rates at every 25th, 30th, 40th, 50th, and 100th frame.

Results demonstrated that:

- GPT-4o-mini could reliably generate relevant commentary for single-ball clips even with sparse input (every 100th frame), confirming that the model can abstract high-level context from limited input cues.
- GPT-4o (full model) provided more robust performance across both single- and multi-event inputs, but required denser sampling (every 30th–50th frame) for multi-event scenarios to maintain temporal alignment and commentary accuracy.

This experiment confirms that frame sampling can be significantly optimized for cost-efficiency in production use cases, without compromising the semantic quality of the generated output.

Part 3 - Google Text to Speech Model

In the first phase of our text-to-speech (TTS) pipeline, we incorporated Google Text-to-Speech (gTTS) to synthesize the cricket commentary generated by the GPT model into audio files. [5]gTTS is a lightweight Python library that leverages Google's cloud-based TTS service and allows for rapid conversion of text into MP3 audio.

The process involved taking the generated commentary text as input, using gTTS to synthesize speech, and saving the output locally. This approach allowed for quick prototyping without the need for GPUs, local model weights, or complex configurations.

The primary advantages of using gTTS were its ease of use, support for multiple languages, and fast turnaround time. However, a key limitation was that the synthesized audio lacked expressive variations such as pitch changes, tone modulation, or emotional delivery.

Despite these limitations, gTTS effectively enabled us to test the coherence, clarity, and contextual relevance of the generated cricket commentary in audio form. For future work, we plan to replace gTTS with a more expressive and natural-sounding TTS model like Bark or ElevenLabs to enhance the overall listening experience.

Results

Part 1 - Event Recognition Model

The event detection model was evaluated qualitatively by inspecting whether each detected segment corresponded to a distinct cricketing event, such as a delivery, replay, or transition. Key observations include:

- The Top-K scoring approach effectively addressed over-segmentation, outperforming simple threshold-based detection.
- Most detected segments aligned well with individual ball deliveries or visually significant moments, such as player reactions or close calls.
- The 3-second minimum gap filter helped suppress false positives in replay-heavy sections of the video.

Overall, the system demonstrated reliable segmentation of longer highlight clips into well-isolated event-based units, which could be directly passed to downstream models for commentary generation or classification.

Part 2 - GPT Vision Model

The GPT-based commentary generation was evaluated qualitatively based on fluency, relevance, and event accuracy.

An ablation study revealed that:

- **GPT-4o-mini** performed reliably for **single-event commentary**, even with sparse sampling (e.g., every 100th frame). However, it struggled to maintain coherence when handling videos with multiple events.
- **GPT-4o** was effective for both **single** and **multi-event** clips. It handled single events well even with sparse frame inputs, but multi-event commentary required

denser sampling (every 30th–50th frame) to preserve context. Reducing frame density below this threshold led to incomplete or incorrect outputs.

Additionally, in cases where the **visual outcome was ambiguous or subtle**, the model occasionally produced fluent but **inaccurate commentary**, highlighting its reliance on frame clarity and distinct event cues.

Due to the lack of existing baselines for vision-based cricket commentary generation, evaluation was performed through **human judgment** focusing on clarity, contextual accuracy, and narrative quality.

Part 3 - Google Text to Speech Model

- The gTTS-generated audio was sufficiently clear, fluent, and understandable for preliminary testing and demonstration purposes.
- The pipeline required minimal computational resources and could be set up quickly, making it ideal for early-stage development.
- However, the lack of expressive features such as dynamic pitch, excitement, or emotional tones made the output sound relatively monotonic compared to real cricket commentators.

Overall, while gTTS fulfilled the needs of initial validation, a more advanced TTS system will be essential to deliver a professional and engaging commentary experience in future iterations of the project.

Conclusion

In this project, we successfully designed and implemented key components required for automated cricket commentary generation. We developed an event recognition model capable of segmenting cricket highlight videos into distinct events, a GPT-4o-based vision-language model that generates short and vivid cricket commentary from visual frames, and a gTTS-based text-to-speech pipeline for synthesizing the generated text into audio commentary. Through ablation studies, we optimized the number of frames needed per event, balancing commentary quality and computational efficiency. The GPT-4o-mini model was found suitable for single-event scenarios, while GPT-4o was more effective for videos containing multiple consecutive events. Evaluation across different videos demonstrated that the generated commentaries were largely accurate, fluent, and contextually appropriate, despite occasional inaccuracies in complex or ambiguous visual scenarios.

For future work, our primary goal is to integrate the three components—event detection, commentary generation, and text-to-speech synthesis—into a single seamless pipeline. This would enable the system to automatically process an input cricket video end-to-end, from detecting events to producing realistic, engaging audio commentary. Further enhancements could include adopting a more expressive TTS model to improve the naturalness of the audio output and fine-tuning the GPT-4o model specifically on cricket-related visual datasets to further improve its domain-specific accuracy.

References

[1] Mervin, *GPT-4 Vision: Adding Football Commentary to Video.*
Available at: https://mer.vin/2023/11/gpt-4-vision-adding-football-commentator-to-video/

[2] OpenAI, *Vision API Documentation.*
Available at: https://platform.openai.com/docs/guides/images?api-mode=chat

[3] OpenAI Help Center, *Using the GPT-4 Vision API.*
Available at: https://help.openai.com/en/articles/8555496-gpt-4-vision-api

[4] OpenAI, *API Platform Overview.*
Available at: https://platform.openai.com/docs/overview

[5] Google Cloud, *Text-to-Speech Documentation.*
Available at: https://cloud.google.com/text-to-speech/docs/

[6] Amazon Web Services, *Revolutionizing Fan Engagement: Generative AI-Powered Live Commentary for the Bundesliga.*
Available at: https://aws.amazon.com/blogs/media/revolutionizing-fan-engagementcer-bundesliga-generative-ai-powered-live-commentary/

[7] CNN, *The Masters: AI Commentary Transforms Golf's Fan Experience.*
Available at: https://www.cnn.com/2023/04/07/golf/the-masters-ai-commentary-spt-intl/index.html

[8] Murata, *Generative AI Technology Revamps Sports Commentary.*
Available at: https://article.murata.com/en-sg/article/generative-ai-technology-revamps-sports-commentary