Automated Cricket Scene Classification using Vision-Language-Models

Karan Sindwani [1], Debasish Mishra [2], Yash Shah[3]

[1]Senior Applied Scientist, Amazon Web Services, Delhi, India
Email: ksindwan@amazon.com | Contact: +91-9971159818

[2] Senior Data Scientist, Amazon Web Services, Bengaluru, India
Email: misdebas@amazon.com | Contact: +91-8008789372

[3]Data Science Manager, Amazon Web Services, Mumbai, India
Email: syash@amazon.com | Contact: +91-9769201634

Abstract

Vision-Language Models (VLMs) have demonstrated impressive capabilities in general-purpose multi-modal tasks, but their adaptation to specialized sports analysis remains relatively unexplored. This paper bridges this gap by investigating VLM's effectiveness for automated cricket scene classification, addressing critical bottlenecks in current workflows that require 45-50 minutes of human intervention. We explore three distinct approaches—zero-shot prompting, few-shot prompting, and Parameter Efficient Fine-Tuning (PEFT) with LoRA—across three fundamental cricket tasks: event marker detection, start of delivery identification, and scoreboard parsing. Our comprehensive experimentation utilizes datasets comprising 30 thousand labeled high-resolution frames spanning 25 matches with balanced distributions across diverse conditions and production styles. Fine-tuned models using PEFT with LoRA achieve 90% accuracy in event marker detection, 98% accuracy in scoreboard parsing, and 95% precision in delivery detection, while requiring significantly less labeled data than traditional approaches. Notably, few-shot prompting approaches achieve competitive performance (84-93% accuracy across tasks) without any training data. Our findings establish a new benchmark for efficiency and accuracy in cricket scene analysis while providing a scalable solution for real-time analysis.

Keywords

Computer Vision, Fine tuning, Scene Classification, Sport Analysis, Vision Language Models

1. Introduction

The evolution of cricket broadcasting and analytics has created an urgent need for efficient, automated video analysis solutions. Current cricket scene understanding systems [1, 2, 3, 14] rely heavily on traditional computer vision approaches combined with substantial manual intervention, typically requiring 45-50 minutes of human processing per game. While these CV-based methods have achieved 75-90% accuracy in prior work, their architectural limitations present fundamental challenges for scaling and generalization. These methods have three key limitations: they rely on fixed feature extractors that cannot leverage semantic reasoning, require extensive task-specific feature engineering, and demand large volumes of labeled data for each new broadcast format or venue condition. Trained analysts must review exceptions and edge cases that arise when models fail to generalize across different match settings, venues, lighting conditions, camera angles, and production styles. This inability to adapt across diverse conditions without retraining has perpetuated the need for manual intervention, creating significant bottlenecks in data extraction and analysis while limiting the feasibility of fully automated systems at scale.

Vision-Language Models (VLMs) [15] offer a promising alternative through their ability to leverage semantic reasoning and contextual understanding. Recent advances in VLMs, particularly models like Claude [16], LLaVA [17], and GPT [18], have shown impressive results in complex visual understanding tasks that require both fine-grained visual analysis and high-level semantic interpretation. Through their transformer-based architectures and large-scale pre-training on diverse image-text pairs, these models demonstrate significant advantages in few-shot learning scenarios and can adapt to specialized domains with minimal labeled data. We investigate Vision-Language Models as an alternative to traditional CV approaches for cricket scene classification. Our work systematically evaluates VLM adaptation strategies from zero-shot to fine-tuning. We establish performance benchmarks and identify the most effective methods for cricket scene classification. Our results demonstrate how VLMs' semantic understanding capabilities address the generalization challenges that have limited prior methods.

Our paper makes the following key contributions:

- We demonstrate the effectiveness of VLMs for domain adaptation across different adaptation strategies: zero-shot prompting achieves 84.5% accuracy without training data, few-shot prompting reaches 84-93% with minimal examples, and fine-tuning achieves up to 98% accuracy. This progression establishes that VLMs can deliver strong performance with significantly reduced labeled data requirements while generalizing robustly across diverse broadcast conditions.

- We show that effective domain adaptation, rather than model scale, is key to achieving state-of-the-art results in specialized tasks. Through systematic analysis of PEFT with LoRA strategies, even smaller models achieve up to 98% accuracy when properly adapted, providing crucial insights for resource-constrained deployment scenarios.

- We present an automated end-to-end solution for cricket scene classification that reduces manual intervention from 45-50 minutes to near-zero per game, addressing the scalability and efficiency limitations of current hybrid systems. Our modular task framework demonstrates clear pathways for extension to related tasks including player tracking, tactical analysis, and action recognition.

2. Literature Survey

Traditional Computer Vision in Cricket Analysis: Cricket scene analysis presents significant computational challenges that distinguish it from conventional sports understanding tasks. The sport's complex multi-actor scenarios, dynamic camera perspectives, and diverse broadcast production standards create substantial barriers for automated analysis systems. Traditional computer vision methodologies have consistently demonstrated limited performance when confronted with these complexities, as evidenced by Kumar et al. [1] who achieved only 50% precision in detecting bowling and batting actions despite using thousands of annotated video frames. The transition toward deep learning architectures has yielded incremental performance gains while simultaneously exposing persistent scalability and cross-domain generalization deficiencies. Contemporary CNN-based systems employing RetinaNet [2] and AlexNet [3] architectures have demonstrated promising capabilities for automated cricket analysis. However, they revealed fundamental architectural limitations. These approaches necessitated extensive dataset preprocessing, exhibited poor cross-domain generalization performance, and required substantial feature engineering to accommodate cricket-specific visual characteristics. The systems remained constrained to fixed camera perspectives and demonstrated an inability to adapt across diverse broadcast formats or venue-specific conditions. Furthermore, Bhat et al. [4] employed exclusively OCR-based information extraction from YOLO models, which proved inadequate for capturing the contextual ordering and spatial relationships of scorecard elements. Similarly, Foysal et al. [5] utilized shallow CNN architectures with grayscale image processing and handcrafted feature extraction, representing a regression to earlier methodological approaches that inherently limit representational capacity and generalization potential.

Domain Adaptation of Vision-Language Models: Adapting VLMs to specialized domains has emerged as a significant research direction. In the medical domain, models such as Med-Flamingo [6] and LLaVA-Med [7] have shown effective adaptation through fine-tuning general-purpose VLMs on biomedical datasets. Similar adaptation strategies have been successfully applied across diverse fields including robotics [8], scientific literature [9], and remote sensing [10]. RT-2 [8] exemplifies this approach by adapting vision-language models for robotic control through training on web-scale multimodal data combined with robotic demonstrations, enabling translation of visual understanding into executable robotic actions. The limited availability of high-quality labeled datasets presents a significant challenge for domain adaptation.

Vision-Language Models in Sports Analysis: Recent developments in Vision-Language Models have shown promising results in sports scene analysis. These models leverage transformer architectures and contrastive learning objectives to align visual and textual representations. Nonaka et al [11]'s work in rugby scene classification showed significant improvements when incorporating VLM outputs compared to pure computer vision approaches, highlighting VLMs' ability to capture complex sports scenarios with minimal labeled data. While [12] proposed combining YOLOv8 with BERT for cricket highlight generation, their architecture was limited to using commentary data for key moment identification. In soccer, domain-adapted VLMs [13] showed a 37.5% boost in video question-answering and markedly better action classification after curriculum-based fine-tuning on sports-specific data.

Despite progress in sports video analysis, critical gaps remain in understanding VLM effectiveness for specialized tasks. No prior work has systematically compared VLM adaptation strategies for cricket scene classification, investigated how model size interacts with adaptation approaches, or demonstrated automated solutions eliminate manual intervention required by current systems. This work addresses these gaps by comprehensively evaluating VLM adaptation strategies across three fundamental cricket tasks, establishing performance benchmarks, and demonstrating robust generalization across diverse broadcast conditions without extensive retraining.

3. Methodology

3.1 Dataset Construction and Annotation

We constructed a comprehensive dataset of 30,000 high-resolution frames (1920×1080 pixels) extracted from 25 cricket matches spanning diverse lighting conditions, camera angles, scorecard and production styles. The dataset exhibits balanced distributions across key variables: lighting conditions include day matches (18,000 frames, 60%) and night matches (12,000 frames, 40%); scorecard opacity varies between high opacity >75% (18,000 frames, 60%) and low opacity <75% (12,000 frames, 40%); scorecard layouts follow runs-first format displaying "Runs-Wickets" (21,000 frames, 70%) and wickets-first format displaying "Wickets/Runs" (9,000 frames, 30%). Frames were strategically sampled at intervals throughout matches to capture different match phases and game situations while avoiding temporal clustering to ensure visual diversity.

Each frame was annotated for all three tasks by experienced cricket analysts, with binary labels provided for event markers and delivery detection, and numerical values (runs, wickets, overs) extracted for scorecard parsing and verified against match footage. Multi-annotator review on a subset of frames achieved inter-annotator agreement of $\kappa > 0.85$ across all tasks, with disagreements resolved through consensus. The dataset was split into 5,000 training frames, 5,000 validation frames, and 20,000 test frames for robust evaluation.

3.2 Task Definitions

Event Marker Detection: This task identifies visual markers and graphical overlays that segment event highlights from live gameplay footage. Positive samples consist of frames containing distinctive event markers such as animated graphics, transition effects, or stylized overlays that precede replay sequences for significant events like wickets, boundaries, or milestones as seen in Figure 1,2. Negative samples include live gameplay footage, replays without markers, crowd shots, advertisements, and field setup sequences. The dataset maintains a balanced class distribution with 51.4% positive and 48.6% negative samples. The negative class deliberately includes "hard negatives" such as replays, crowd shots, and advertisements. These hard negatives share visual similarities with marked events but must be correctly rejected. This ensures models learn discriminative features rather than exploiting spurious correlations.

Start of Delivery Detection: This task identifies the precise moment when a bowler begins their approach to the crease before releasing the ball. Positive samples are defined as frames where the bowler is entering the delivery stride with forward momentum toward the crease, captured from camera angles showing the bowler's run-up (Figure 3). Negative samples strategically include challenging cases such as pre-delivery rituals (field adjustments, bowler walking back to their mark), post-delivery follow-through, and fielder movements to enforce temporal precision and prevent models from triggering on visually similar but temporally incorrect sequences. The dataset maintains a natural class imbalance with approximately 20% positive and 80% negative samples, reflecting realistic match conditions where delivery sequences are interspersed with longer periods of field setup, player discussions, and other match activities.

Scorecard Parsing: This task extracts structured numerical information (runs, wickets, overs) from graphical overlays displaying real-time match statistics. Frames contain visible scorecard overlays with varying opacity levels, diverse backgrounds ranging from static stadium views to moving crowds behind transparent scoreboards and overlapping graphics such as sponsor logos and player name displays. Unlike pure OCR, this task requires understanding spatial relationships to determine which number represents runs versus wickets versus overs based on position and separator characters. The model must adapt to variable layouts where different productions use different ordering conventions ("runs/wickets" in Figure 4 versus "wickets/runs" in Figure 3). It must disambiguate multiple numerical values (runs, wickets, overs, run rate, target score) based on visual context and typical cricket scoring patterns. Additionally, it must segment relevant information from overlapping graphics and visual clutter while maintaining accuracy across dynamic value ranges that change throughout the match.

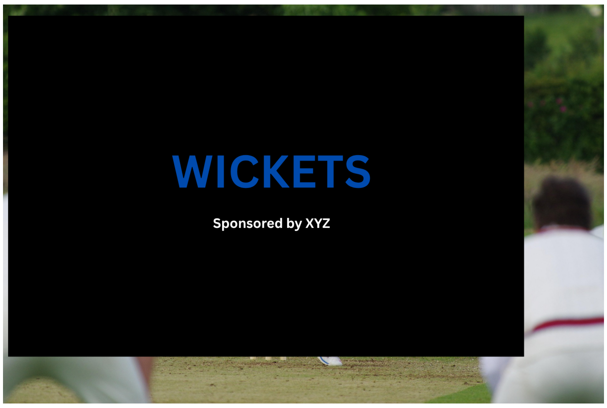

Figure 1: Event Marker for wicket

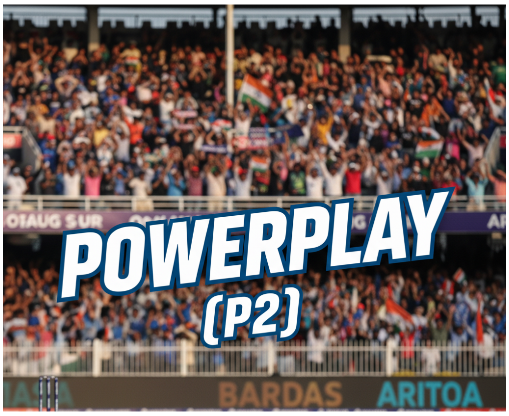

Figure 2 : Event marker for powerplay

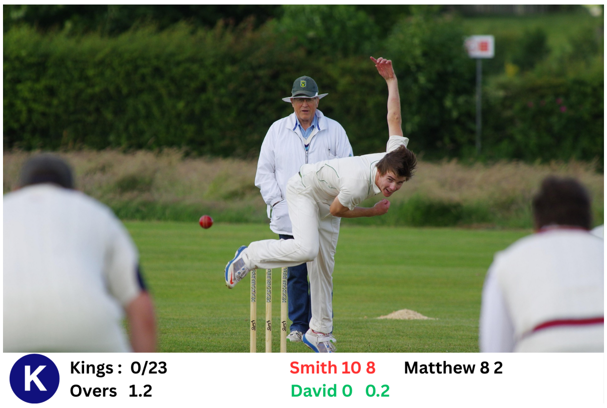

Figure 3 : Start of delivery and scorecard with wickets/runs

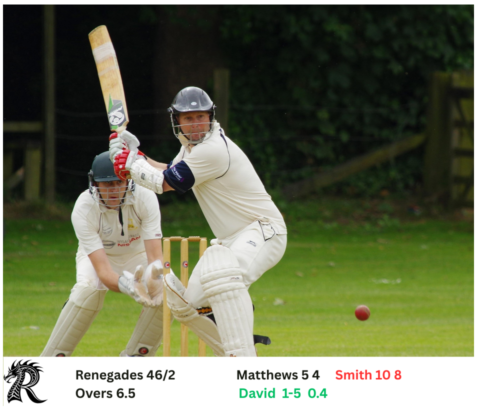

Figure 4: Score card with runs/wickets

3.3 Experimental Approaches

We investigate three distinct paradigms for adapting Vision-Language Models to cricket scene classification, each representing different points on the data-efficiency and performance spectrum.

Zero-Shot Prompting: Zero-shot prompting leverages pre-trained VLMs' visual reasoning capabilities without any task-specific training. We utilize Claude 3 models (Haiku and Sonnet variants) through Amazon Bedrock, designing carefully crafted prompts that provide task context, describe visual characteristics to identify, and specify output format requirements. For event marker and delivery detection, prompts include detailed descriptions of target visual patterns and request binary classification outputs (see Figure 5). For scorecard parsing, prompts specify the numerical fields to extract and their expected formats (see Figure 6). This approach establishes baseline performance achievable with general-purpose VLMs and demonstrates their out-of-the-box capabilities for specialized sports analysis tasks.

Few-Shot Prompting: Few-shot prompting extends zero-shot approaches by incorporating a small number of labeled examples directly in the prompt context. For each task, we provide three carefully selected example pairs (six total examples) that showcase diverse scenarios including different production styles, lighting conditions, and visual variations. Examples are embedded as image-text pairs within the prompt, demonstrating both positive and negative cases to enhance discrimination of subtle visual patterns (Figures 5 and 6). This approach evaluates VLMs' ability to rapidly adapt to domain-specific patterns through in-context learning with minimal labeled data.

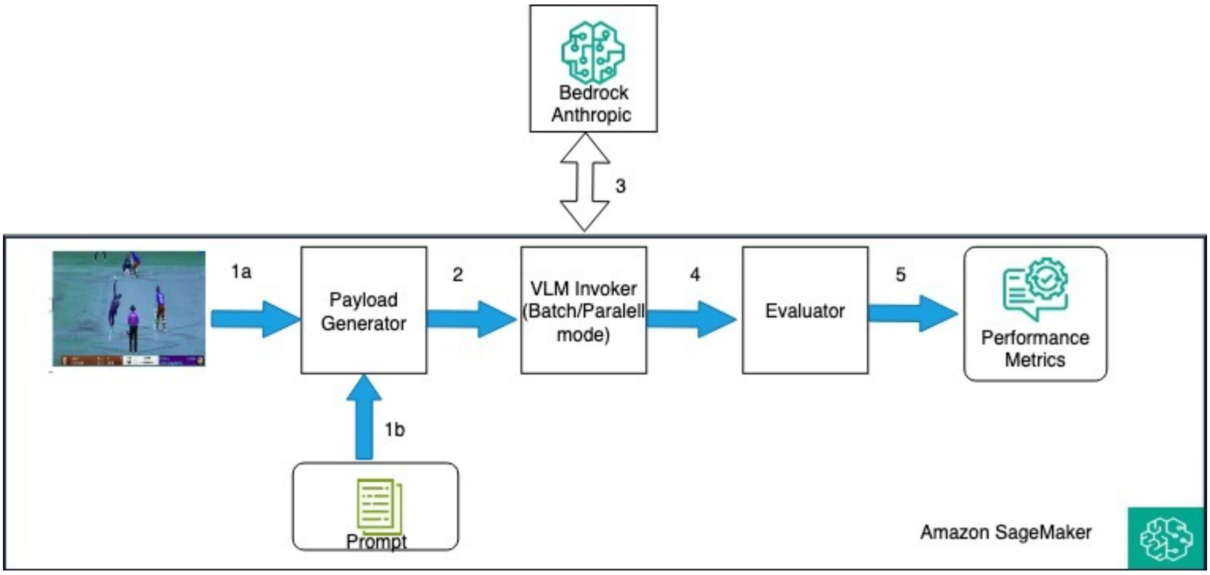

Figure 5 : Prompting workflow for start of delivery and event detection

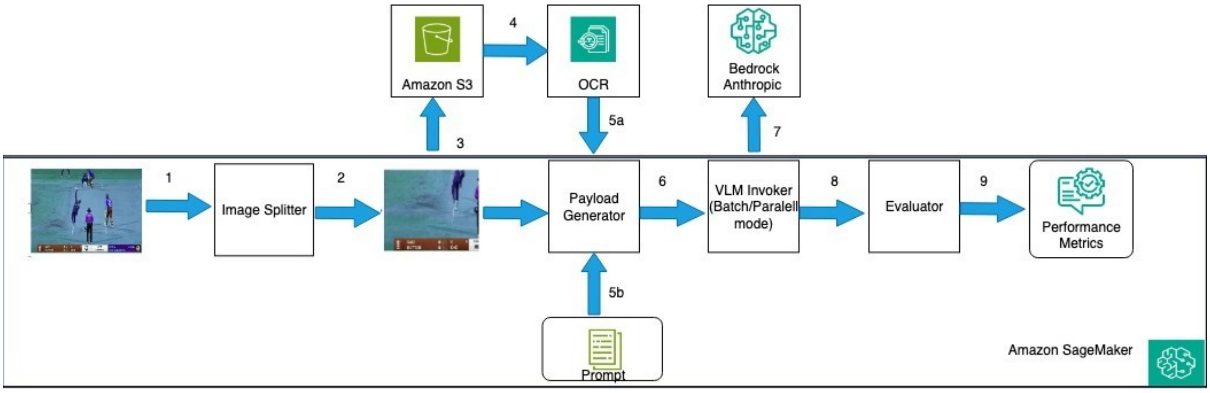

Figure 6: Prompting workflow for Scorecard parsing

Parameter Efficient Fine-Tuning with LoRA: Fine-tuning with Low-Rank Adaptation (LoRA) enables domain-specific model adaptation while maintaining computational efficiency by introducing trainable low-rank decomposition matrices into model layers. We fine-tune two open-source VLMs: Qwen2-VL-7B-Instruct (7 billion parameters) and SmolVLM-Instruct (smaller, resource-efficient model) on AWS g5.12xlarge instances with NVIDIA A10G GPUs (see Figure 7 for workflow). All experiments use batch size 16, learning rate 2×10^{-4} with cosine annealing scheduler, 2 training epochs with early stopping based on validation loss, and LoRA rank r=8.

All experiments use 5,000 training frames with 5,000 validation frames and 20,000 test frames, batch size 16, learning rate 2×10^{-4} with cosine annealing scheduler, 2 training epochs with early stopping based on validation loss, and LoRA rank r=8. To systematically evaluate the impact of different training strategies, we conduct an ablation study exploring three LoRA adaptation approaches:

- attention-only, applying LoRA exclusively to attention layers,
- comprehensive, applying LoRA to attention layers, fully connected layers, and language model head,
- all-linear, applying LoRA to all linear layers in the model.

This ablation design allows us to isolate the effects of layer adaptation scope on model performance across different cricket scene classification tasks.

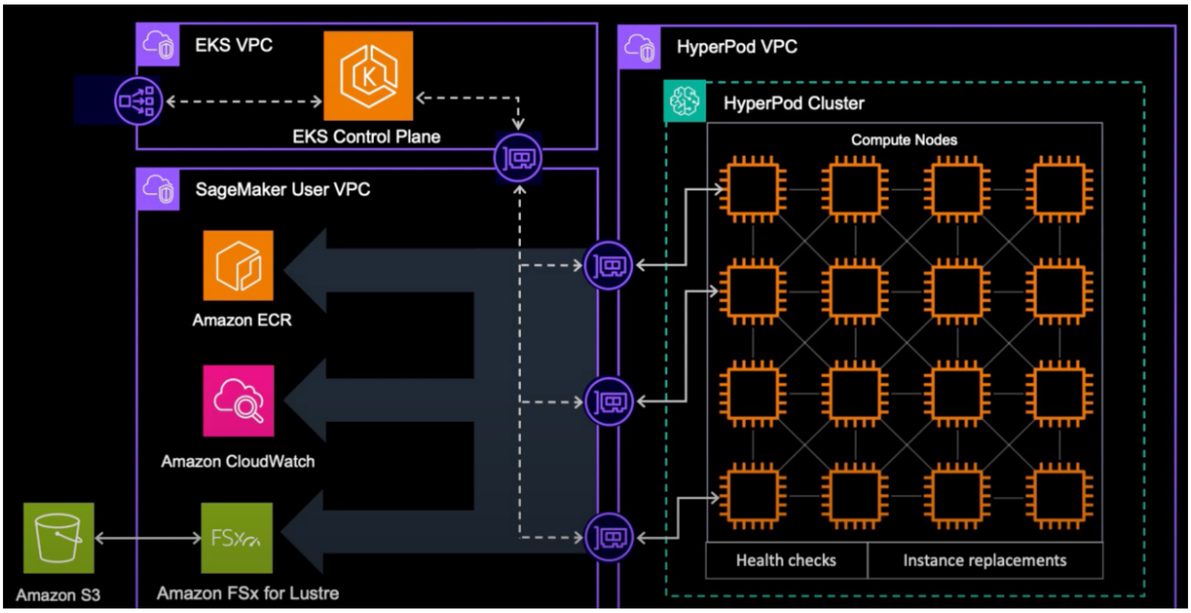

Figure 7 : Finetuning workflow

4. Results and Discussion

We evaluated Vision-Language Models across three cricket scene classification tasks using zero-shot prompting, few-shot prompting, and parameter-efficient fine-tuning approaches. Our experiments demonstrated that VLMs achieved strong performance across all tasks, with fine-tuned models reaching up to 98% accuracy while prompting-based approaches provided competitive results without requiring training data. The following subsections present detailed results for each task, followed by failure case analysis

4.1 Event Marker Detection

Event marker detection experiments utilized a balanced dataset comprising 51.4% marker-present and 48.6% marker-absent frames, with results summarized in Table 1. Zero-shot prompting established strong baselines, with Claude Haiku achieving 84.5% accuracy, demonstrating that pre-trained VLMs possess substantial visual reasoning capabilities for this task without any domain-specific training. Interestingly, Claude Sonnet underperformed at 78% accuracy despite being a larger model, suggesting that additional model capacity does not automatically translate to better performance on straightforward visual discrimination tasks. Few-shot prompting provided minimal gains, with Claude Haiku improving only to 84.6% accuracy, indicating that the task's visual patterns are sufficiently captured by pre-training and additional examples offer limited value.

Fine-tuning revealed a critical insight: model size and adaptation strategy interact in complex ways that determine performance outcomes. Qwen2-7B demonstrated robustness across all three LoRA strategies, with performance ranging narrowly from 88% to 90% accuracy. This consistency suggested that larger models can effectively leverage different adaptation approaches, with attention layers capturing most discriminative features (88% accuracy) and broader adaptation providing modest incremental gains (90% accuracy with all-linear). The slight improvements indicated diminishing returns from more extensive layer adaptation for this relatively straightforward classification task.

SmolVLM's results revealed a different story, revealing that smaller models require careful adaptation strategy selection. All-linear adaptation achieved 88% accuracy, matching Qwen2-7B's attention-only performance and demonstrating that smaller models can compete with larger ones when properly configured. However, comprehensive layer training catastrophically failed at 72% accuracy with 46% recall, representing a 34% recall collapse compared to the 80% recall achieved by other strategies. The failure mode showed extremely high precision (98%) coupled with severely degraded recall indicating that the model learned to minimize loss by defaulting to negative predictions rather than learning discriminative features. This suggested that adapting intermediate layers in smaller models can destabilize training, particularly when the model lacks sufficient capacity to effectively utilize the additional trainable parameters. These findings established that effective VLM deployment requires matching adaptation scope to model capacity, with smaller models benefiting from either focused (attention-only) or comprehensive (all-linear) adaptation while avoiding intermediate strategies that may induce training instability.

Table 1 : Experimental Results for event marker detection

Technique	Model	Accuracy	Precision	Recall
Zero-shot	Claude Haiku	84.5	84	86.9
Zero-shot	Claude Sonnet	78	86	70
Few-shot	Claude Haiku	84.6	85	85
Few-shot	Claude Sonnet	80	82	78
Fine tune (5k) *	Qwen2-7B	90	95	85
Fine tune (5k) †	Qwen2-7B	89	92	86
Fine tune (5k) ^	Qwen2-7B	88	92	86
Fine tune (5k) *	SmolVLM	88	95	80
Fine tune (5k) †	SmolVLM	72	98	46
Fine tune (5k) ^	SmolVLM	84	90	80

* adapters for all linear layers

† adapters for attention layers, fc layers and LM head

^ adapters for only attention layers

4.2 Scorecard Parsing

Scorecard parsing experiments focused on extracting structured numerical information (runs, wickets, overs) from graphical overlays, with results presented in Table 2. We processed only the bottom-left quarter of each frame where scorecard overlays are consistently positioned, reducing computational overhead by 75% while maintaining access to relevant information. Zero-shot prompting achieved modest performance, with Claude models reaching 55-71% accuracy across the three fields, demonstrating limited capability to interpret structured numerical information without examples. The relatively poor performance, particularly for wickets at 55-61% accuracy, highlighted that understanding spatial relationships and semantic meaning of numerical values requires task-specific context that pre-training alone does not provide.

Few-shot prompting transformed performance dramatically, with accuracy improvements of 27-54% across all fields. Claude Haiku achieved 88.5% for runs, 84.5% for wickets, and 92.6% for overs, while Claude Sonnet reached comparable levels at 88-93%. The particularly large improvements for wickets and overs compared to runs revealed an important pattern: fields requiring more complex spatial reasoning benefit disproportionately from contextual examples. Runs, being typically the first and most prominent number, are easier to identify even without examples, while wickets and overs require understanding positional context and separator conventions that few-shot examples effectively demonstrate. These results substantially exceeded traditional OCR-based approaches, which struggle with variable layouts and overlapping graphics, validating that scorecard parsing requires semantic understanding beyond character recognition.

Fine-tuning experiments revealed that task complexity determines optimal adaptation strategy. Attention-only training achieved strong performance for runs (94%) and wickets (96%) but failed dramatically on overs extraction at only 76% accuracy, representing an 18% gap. This disparity indicated that overs extraction requires understanding more complex spatial relationships and contextual patterns that attention mechanisms alone cannot capture. Comprehensive layer training addressed this limitation, improving overs accuracy to 91%, while all-linear adaptation achieved the best overall results at 93% for overs. The critical insight here is that unlike event marker detection where visual discrimination happens primarily in attention layers, scorecard parsing requires the fully connected and output layers to learn structured extraction patterns, spatial relationship encoding, and disambiguation logic for multiple numerical fields. The 17% improvement in overs accuracy when moving from attention-only to all-linear adaptation demonstrated that task characteristics not just dataset size determine whether comprehensive layer adaptation justifies its additional computational cost.

Table 2: Experimental Results for scorecard parsing

Technique	Model	Runs	Wickets	Overs
Zero-shot	Claude Haiku	70	55	60
Zero-shot	Claude Sonnet	68	61	71
Few-shot	Claude Haiku	88.5	84.5	92.6
Few-shot	Claude Sonnet	88	86	93
Fine tune (5k) *	Qwen2-7B	96	98	93
Fine tune (5k) †	Qwen2-7B	97	98	91
Fine tune (5k) ^	Qwen2-7B	94	96	76

* adapters for all linear layers

† adapters for attention layers, fc layers and LM head

^ adapters for only attention layers

4.3 Start of Delivery Detection

Start of delivery detection experiments utilized an imbalanced dataset where positive sequences comprised approximately 20% of total frames, necessitating precision and recall as primary evaluation metrics, with results presented in Table 3. Zero-shot prompting revealed the fundamental challenge of temporal action recognition in imbalanced settings. Claude Haiku achieved high recall at 95% but poor precision at only 54%, indicating that the model over-identified positive sequences and generated substantial false positives by triggering on visually similar movements such as fielders running or bowlers walking back. Claude Sonnet offered more balanced performance at 64% precision and 61% recall, but both models demonstrated that general-purpose VLMs struggle with the fine-grained temporal discrimination required to distinguish delivery strides from other bowling-related movements without domain-specific training.

Few-shot prompting produced an unexpected precision-recall tradeoff that actually degraded overall performance. Claude Haiku improved precision to 78% but recall collapsed to 56%, while Claude Sonnet achieved 80% precision but recall plummeted to only 31%. This pattern suggested that providing examples of challenging negative cases made models overly conservative, causing them to miss genuine delivery sequences to avoid false positives. The dramatic recall reduction in Claude Sonnet from 61% to 31% indicated that few-shot learning can be counterproductive for highly imbalanced temporal detection tasks, where the model learns to err on the side of caution rather than developing robust discriminative capabilities. This finding highlighted a critical limitation of prompting-based approaches for complex temporal tasks with severe class imbalance.

Fine-tuning demonstrated substantial and progressive improvements that validated the value of domain-specific adaptation for this challenging task. Attention-only training achieved balanced performance at 86% precision and 85% recall, representing significant gains over prompting approaches and establishing that learned temporal features substantially outperform in-context learning for this task. Comprehensive layer training improved results to 89% precision and 90% recall, while all-linear adaptation achieved near-optimal performance at 95% precision and 98% recall. The progressive improvement from attention-only to comprehensive to all-linear revealed that start of delivery detection benefits substantially from extensive layer adaptation, likely because the task requires complex temporal reasoning to distinguish subtle differences in bowler movements and fine-grained visual discrimination to identify the precise moment of delivery stride initiation. Unlike event marker detection where attention layers sufficed, this temporal detection task required the model to learn sophisticated motion patterns and temporal dependencies that only emerge when fully connected and output layers are adapted.

Table 3 : Experimental Results Start of Delivery Detection

Technique	Model	Precision	Recall
Zero-shot	Claude Haiku	54	95
Zero-shot	Claude Sonnet	64	61
Few-shot	Claude Haiku	78	56
Few-shot	Claude Sonnet	80	31
Fine tune (5k) *	Qwen2-7B	95	98
Fine tune (5k) †	Qwen2-7B	89	90
Fine tune (5k) ^	Qwen2-7B	96	85

* adapters for all linear layers

† adapters for attention layers, fc layers and LM head

^ adapters for only attention layers

4.4 Failure Case Analysis

For event marker detection, zero-shot and few-shot models occasionally misclassified replays without markers as positive samples when they contained slow-motion effects or dramatic camera angles, indicating reliance on stylistic cues rather than explicit marker presence. Fine-tuned models reduced these errors but failed on edge cases with unusual transparency levels or partial occlusion. The SmolVLM comprehensive adaptation failure manifested as systematic rejection of positive samples, with the model defaulting to negative predictions to achieve high precision (98%) while missing nearly half of true positives (46% recall).

For scorecard parsing, all models struggled with low-opacity overlays against complex moving backgrounds, with zero-shot approaches achieving only 55-60% accuracy on such frames compared to 88-93% on clear overlays. The attention-only fine-tuning weakness in overs extraction (76% accuracy) occurred when layouts deviated from common patterns, revealing that attention mechanisms could identify individual numbers but failed to understand structural relationships determining which number represents overs versus runs or wickets.

For start of delivery detection, zero-shot/few models triggered false positives on fielders running, umpires signaling, or bowlers walking back. Fine-tuned models substantially reduced both error types but occasionally failed on unusual camera angles or when bowler approaches were partially occluded by on-screen graphics.

5. Conclusion

This work investigated Vision-Language Models for automated cricket scene classification across three fundamental tasks: event marker detection, start of delivery identification, and scorecard parsing. Our systematic evaluation of zero-shot prompting, few-shot prompting, and parameter-efficient fine-tuning with LoRA demonstrated that VLMs achieved strong performance across all tasks, with fine-tuned models reaching 90% accuracy for event markers, 98% for scorecard parsing, and 95% precision for delivery detection. Notably, few-shot prompting achieved competitive performance (85-93% accuracy) without any training data, establishing VLMs as viable solutions for scenarios where extensive labeled datasets are unavailable or cost-prohibitive.

Our experiments revealed that task complexity and model capacity interact in critical ways that determine optimal adaptation strategies. Event marker detection, being primarily visual discrimination, achieved strong results with attention-only adaptation, while scorecard parsing and start of delivery detection required comprehensive layer adaptation due to their demands for spatial reasoning and temporal understanding. The catastrophic failure of

SmolVLM with comprehensive adaptation (72% accuracy, 46% recall) contrasted with its success using all-linear adaptation (88% accuracy) demonstrated that smaller models require careful strategy selection, with intermediate adaptation approaches potentially inducing training instability.

These findings address the critical bottleneck in current cricket analysis systems that require 45-50 minutes of manual intervention per game. Our automated pipeline reduces this to near-zero while maintaining high accuracy across diverse broadcast conditions, lighting scenarios, and production styles. The modular three-task framework provides a foundation for comprehensive cricket analytics, with clear pathways for extension to related tasks including player tracking, tactical analysis, and action recognition. Future work should explore temporal modeling through video transformers to capture sequential patterns, multi-modal integration combining visual analysis with audio commentary, and expansion to additional cricket analysis tasks to create comprehensive automated systems for sports video understanding.

References

[1] A. Kumar, J. Garg, and A. Mukerjee, “Cricket activity detection,” in *International Image Processing, Applications and Systems Conference*, IEEE, Nov. 2014, pp. 1–6. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1109/ipas.2014.7043264

[2] T. Moodley and D. van der Haar, “Cricket Scene Analysis Using the RetinaNet Architecture,” in *Lecture Notes in Computer Science*, Cham: Springer International Publishing, 2021, pp. 197–206. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1007/978-3-030-93420-0_19

[3] T. Moodley and D. van der Haar, “Scene Recognition Using AlexNet to Recognize Significant Events Within Cricket Game Footage,” in *Lecture Notes in Computer Science*, Cham: Springer International Publishing, 2020, pp. 98–109. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1007/978-3-030-59006-2_9

[4] R. S. Bhat, J. O, P. P. P, P. Kumar Vedurumudi, and D. K. N, “Cricket Video Summarization Using Deep Learning,” in *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, IEEE, Apr. 2023, pp. 1–6. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1109/i2ct57861.2023.10126359

[5] Md. F. A. Foysal, M. S. Islam, A. Karim, and N. Neehal, “Shot-Net: A Convolutional Neural Network for Classifying Different Cricket Shots,” in *Communications in Computer and Information Science*, Singapore: Springer Singapore, 2019, pp. 111–120. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1007/978-981-13-9181-1_10

[6] M. Moor *et al.*, “Med-Flamingo: a Multimodal Medical Few-shot Learner,” arXiv.org. [Online]. Available: https://arxiv.org/abs/2307.15189

[7] C. Li *et al.*, “LLaVA-Med: Training a Large Language-and-Vision Assistant for Biomedicine in One Day,” arXiv.org. [Online]. Available: https://arxiv.org/abs/2306.00890

[8] A. Brohan *et al.*, “RT-2: Vision-Language-Action Models Transfer Web Knowledge to Robotic Control,” arXiv.org. [Online]. Available: https://arxiv.org/abs/2307.15818

[9] X. Li, Y. Sun, W. Cheng, Y. Zhu, and H. Chen, “Chain-of-region: Visual Language Models Need Details for Diagram Analysis.” [Online]. Available: https://iclr.cc/virtual/2025/poster/29954?

[10] J. D. Silva, J. Magalhães, D. Tuia, and B. Martins, “Multilingual Vision-Language Pre-training for the Remote Sensing Domain,” in *Proceedings of the 32nd ACM International Conference on Advances in Geographic Information Systems*, New York, NY, USA: ACM,

Oct. 2024, pp. 220–232. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1145/3678717.3691318

[11] N. Nonaka, R. Fujihira, T. Koshiba, A. Maeda, and J. Seita, "Rugby Scene Classification Enhanced by Vision Language Model," in *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Jun. 2024, pp. 3256–3266. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1109/cvprw63382.2024.00331

[12] H. Sattar, M. S. Umar, E. Ijaz, and M. U. Arshad, "Multi-Modal Architecture for Cricket Highlights Generation: Using Computer Vision and Large Language Model," in *2023 17th International Conference on Open Source Systems and Technologies (ICOSST)*, IEEE, Dec. 2023, pp. 1–6. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1109/icosst60641.2023.10414235

[13] T. Jiang, H. Wang, M. S. Salekin, P. Atighehchian, and S. Zhang, "Domain Adaptation of VLM for Soccer Video Understanding," in *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Jun. 2025, pp. 6101–6111. Accessed: Nov. 18, 2025. [Online]. Available: https://doi.org/10.1109/cvprw67362.2025.00608

[14] A. Gupta and S. B. M, "Cricket stroke extraction: Towards creation of a large-scale cricket actions dataset," arXiv.org. [Online]. Available: https://arxiv.org/abs/1901.03107

[15] F. Bordes *et al.*, "An Introduction to Vision-Language Modeling," arXiv.org. [Online]. Available: https://arxiv.org/abs/2405.17247

[16] "Home \ Anthropic." Accessed: Nov. 18, 2025. [Online]. Available: https://www.anthropic.com/

[17] haotian-liu, "GitHub - haotian-liu/LLaVA: [NeurIPS'23 Oral] Visual Instruction Tuning (LLaVA) built towards GPT-4V level capabilities and beyond.," GitHub. Accessed: Nov. 18, 2025. [Online]. Available: https://github.com/haotian-liu/LLaVA

[18] Chatgpt. Accessed: Oct. 31, 2025. [Online]. Available: https://openai.com/

6. Appendix

6.1 Prompt templates

Event Marker Detection

```
You are a helpful AI assistant, help as much as you can.

Please analyze the attached image of a livestream of a cricket match.
Your job is to determine if the image consists of a replay graphic or not.  It may be
represented by a logo of a cricket league, a keyword four or a keyword six.

Please provide your findings in the following structured JSON format:
{
'graphic' : 1 if a graphic is present, 0 if the image represent a cricket game, blurred view of
a game or
anything else on a cricket ground.
'explanation' : "State the reason"  }

Return only the JSON and nothing else.
```

Start of delivery

```
1. You are an expert image analyst for cricket matches. Given an image from a cricket match,
your task is to identify if the image contains any of the following scenes given in the <scenes>
tag

<scenes> A. Image shows the wicket and Bowler standing with the ball B. Image shows the wicket
and Bowler running or walking towards the wicket C. Image shows the wicket and Bowler performing
the delivery action on the wicket to deliver the ball to the batsman D. Image shows the wicket
and Bowler delivered the ball to the batsman and doing a follow-through run E. Image shows the
wicket and Batsman looking front towards the baller, and swinging the bat to hit the ball F.
Image shows the wicket and Batsman looking front towards the baller and hits the ball with the
bat and does a follow-through </scenes>

Your answer should be Runup:1 if the image satisfies all the criteria in <scenes> tag otherwise
Runup:0 if none of the criteria in <scenes> tag is fulfilled

Please provide your findings in the following structured JSON format:
{‘Runup’ : 1 , If the image contains any of the scenes given in <scenes> tag, 0 if image does
not contains any of the scenes given in <scenes> tag 'explanation' : "explain your reasoning for
identifying the specific scene or concluding that none of the given scenes are present. Your
response should be concise and focused on the task at hand.”
}

Return only the JSON and nothing else.
```

Scorecard

```
1. You are an expert in extracting relevant information from a scorecard content of a cricket
match
Given below inside <scorecard_text> tags is the scorecard content in text from a cricket match.
Follow the instructions given inside <instructions> tag to do your job <scorecard_text>
{text}
<scorecard_text>

The runs scored at the moment, wickets gone and total overs bowled at the moment can be present
in different formats as given inside <runs_wicket_format> tags <runs_wicket_format> Format1:
Runs-Wickets, example : 130-2, it means 130 runs have been scored at the moment with the fall of
2 wickets  Format2: Wickets/Runs, example: 1/3, it means 1 wicket is lost and 3 runs are scored
at the moment </runs_wicket_format>

Total overs bowled can be present in formats as given inside <overs_format> tag
<overs_format>
Each over is represented by a number. For example, the first over of the innings is referred to
as "1st over," the second over is "2nd over," and so on.
Overs can be represented using a decimal notation. Each full over is denoted by a whole number,
while the additional deliveries (if any) are represented as a fraction after the decimal point.
For example, "6.3" means six full overs plus three additional deliveries have been bowled
</overs_format>

<examples> <example1> scorecard - ['3/93','11.3', 'RENSHAW 29 (22)','BILLINGS 23
(19)','SUTHERLAND 0/15(2.3)']
analysis - The format of the runs and wickets is in the format "Wickets/Runs". Hence runs scored
at the moment is 93 and wickets gone is 3. Overs bowled is in decimal format which is 11.3
Hence, runs - 93, wickets - 3, overs - 11.3 </example1> <example2>
scorecard - ['MAYERS','9 3','31-0','CG', '2.1', '|','UNITED','THIS OVER','1']
analysis - The format of the runs and wickets is in the format "Runs-Wickets". Hence runs scored
at the moment is 31 and wickets gone is 0. Total Overs bowled is in decimal format which is 2.1
Hence runs - 31, wickets - 0, overs - 2.1
</example2> <example3> scorecard - ['HURRICANES','P','TO WIN: 85 RUNS OFF 34 BALLS AT 15.00
RPO','STR 3/164 (20)','9/80','14.2 ISMAIL 3 (8)','GIBSON 3 (7)','ADAMS 0/6 (2.2)']
analysis - The format of the runs and wickets is in the format "Wickets/Runs". Hence runs scored
at the moment is 80 and wickets gone is 9. Total Overs bowled is in decimal format which is 14.2
Hence, runs - 80, wickets - 9, overs - 14.2 </examples>

<instructions>
1.You have to extract the runs scored at the moment by the batting team in the current innings,
wickets gone in the current innings and total overs bowled in the current innings from the
content in <scorecard_text> tags
2.Follow the guidelines given inside <runs_wicket_format> tag to understand how to interpret
runs and wickets from the content. Also follow the guidelines given inside <overs_format> tag to
understand how to interpret overs bowled from the content
3.Learn from the examples given inside <examples> tag to understand how the runs, wickets and
overs are extracted from the scorecard
4.Think step by step in <analysis> tag before generating the answer. Focus on the innings of the
current batting team to extract the required information
5.Return your findings in the following format:
<summary>
"runs": [number, "confidence_level"],
"wickets": [number, "confidence_level"],
"over": [number, "confidence_level"],
</summary> <analysis> "analysis": "Detailed analysis - findings, and summary" </analysis>
</instructions>
```