Concept Paper

Cricket Stroke Classification Using Temporal Deep Learning Models: A Systematic Comparative Study of Sequence Modeling Architectures for Fine-Grained Action Recognition

Abhinav Bansal * and Shally Vats

Posted Date: 28 April 2026

doi: 10.20944/preprints202604.1970.v1

Keywords: cricket stroke classification; action recognition; temporal deep learning; LSTM; BiLSTM; transformer; computer vision; sports video analysis; sequence modeling; sports video analysis; ResNet features; McNemar's test; confusion analysis

 doi:10.20944/preprints202604.1970.v1

Concept Paper

Cricket Stroke Classification Using Temporal Deep Learning Models: A Systematic Comparative Study of Sequence Modeling Architectures for Fine-Grained Action Recognition

Abhinav Bansal * and Shally Vats

Department of Information Technology Manipal University Jaipur, Jaipur, Rajasthan, 303007, India

* Correspondence: bansalabhinav.1910@gmail.com

Abstract

Fine-grained action recognition in sports video analysis presents a constellation of challenges that distinguish it from conventional human activity recognition tasks. When actions share substantial visual and temporal overlap—as is the case with cricket batting strokes—the discriminative cues necessary for accurate classification reside in subtle kinematic variations that operate at fine spatial and temporal granularities. This paper presents a comprehensive empirical investigation into the efficacy of contemporary temporal deep learning architectures for the classification of fifteen distinct cricket batting strokes using the *CricShot10k*dataset, a curated collection of approximately 10,000 video clips with balanced class representation. We conduct a rigorously controlled comparative evaluation of four distinct modeling paradigms: a non-temporal convolutional baseline employing frame-averaged ResNet50 features, a unidirectional Long Short-Term Memory (LSTM) network, a bidirectional LSTM (BiLSTM) architecture, and a Transformer encoder with multi-head self-attention. All models are trained and evaluated under identical experimental conditions, including uniform frame sampling (32 frames per video), consistent spatial feature extraction (2048-dimensional ResNet50 embeddings), matched data splits, and standardized optimization hyperparameters. This method-ological rigor ensures that observed performance differences can be attributed cleanly to architectural choices in temporal modeling rather than confounding experimental variables. Our empirical results establish a clear performance hierarchy among the evaluated architectures. The BiLSTM model achieves the highest classification accuracy at 44.0 %, followed by the unidirectional LSTM at 42.0 %, the non-temporal baseline at 34.0 %, and notably, the Transformer architecture at only 23.0 %—a result that falls substantially below even the frame-averaged baseline. Statistical validation through McNemar's test confirms that the 2 % improvement from LSTM to BiLSTM is highly significant ($p < 0.001$), indicating systematic correction of specific misclassification patterns rather than random variation. Detailed per-class analysis and confusion matrix examination reveal pronounced performance heterogeneity across the fifteen stroke categories. High-performing classes such as the pull shot and cut shot exhibit F1-scores exceeding 0.65, attributable to their distinctive lateral motion patterns that create separable trajectories in the spatiotemporal feature space. Conversely, a cluster of front-foot strokes—including the cover drive, defensive shot, down-the-wicket stroke, and lofted offside drive—constitute the primary locus of classification error, with pairwise confusion rates frequently exceeding 40 %. We attribute these persistent confusions to the visual and temporal similarity of these strokes, which share common initial footwork, comparable bat trajectories through the downswing phase, and overlapping follow-through dynamics. The frame-level ResNet50 features, while capturing high-level spatial semantics effectively, lack the temporal granularity and explicit motion encoding necessary to resolve these fine-grained distinctions. Our error analysis further reveals that the model struggles to capture subtle kinematic cues such as bat face angle at impact, degree of wrist rotation, bat elevation trajectory, and the precise timing of weight transfer—features that human experts rely upon for stroke differentiation. The

 doi:10.20944/preprints202604.1970.v1

pronounced underperformance of the Transformer architecture provides a critical case study in the data efficiency limitations of attention- based models for fine-grained visual sequence tasks. Despite its theoretical capacity to model long-range dependencies without the sequential bottleneck of recurrence, the Transformer's lack of built-in inductive biases regarding temporal locality and sequen- tial ordering renders it vulnerable to overfitting in moderate- data regimes. With approximately 400 training examples per class, the *CricShot10k*dataset provides insufficient statistical signal for the Transformer to learn robust spatiotemporal attention patterns from scratch, resulting in degraded generalization performance. This finding underscores an important principle for practitioners: architectural sophistication does not guarantee empirical superiority; rather, the choice of temporal modeling strategy must be carefully aligned with dataset scale and task characteristics. The contributions of this work are fourfold. First, we establish a rigorous and reproducible benchmark for cricket stroke classification on the *CricShot10k*dataset, providing a standardized reference point for future research. Second, through systematic architectural comparison, we quantify the marginal contribution of bidirectional temporal context to classification performance and demonstrate its statistical significance. Third, we provide granular error analysis that identifies specific stroke categories and confusion pairs that dominate classification failures, thereby directing future research efforts toward the most impactful areas for improvement. Fourth, we offer a critical assessment of Transformer limitations in fine-grained, moderate-data video classification scenarios, contributing to the broader understanding of attention mechanism applicability. This paper is organized as follows. Section II reviews related work in action recognition, temporal modeling, and sports video analysis, situating our contribution within the broader research landscape. Section III details our methodological framework, including dataset characteristics, preprocessing pipelines, feature extraction protocols, and architectural specifications. Section IV presents comprehensive experimental results encompassing aggregate performance metrics, training dynamics, statistical validation, and per-class analysis. Section V provides extended discussion of key findings, interpreting the relative performance of architectures and analyzing the structure of classification errors. Section VI addresses limitations and outlines promising directions for future investigation. Section VII concludes with a summary of contributions and their implications for the field.

Keywords: cricket stroke classification; action recognition; temporal deep learning; LSTM; BiLSTM; transformer; computer vision; sports video analysis; sequence modeling; sports video analysis; ResNet features; McNemar's test; confusion analysis

---

I. Introduction

A. The Challenge of Fine-Grained Sports Action Recognition

The automated analysis of human movement from video data has emerged as one of the most active and impactful research domains within computer vision and machine learning. From surveillance and security applications to healthcare monitoring and human-computer interaction, the ability to algorithmically interpret human actions enables a vast array of technological capabilities. Within this broad landscape, sports video analysis occupies a distinctive and particularly demanding niche. Unlike generic action recognition scenarios where the objective is to distinguish semantically distinct activity categories (e.g., walking versus jumping versus throwing), sports action recognition typically operates in a fine-grained regime where the classes of interest share extensive visual, spatial, and temporal commonality.

Consider the task that motivates this paper: the classification of cricket batting strokes from video footage. The fifteen stroke categories represented in standard cricket coaching taxonomies—including the cover drive, straight drive, on drive, pull shot, hook shot, cut shot, sweep shot, defensive stroke, and various lofted variations—constitute a graded continuum of motion patterns rather than discrete, easily separable action categories. A cover drive and an off drive share the same

 doi:10.20944/preprints202604.1970.v1

fundamental biomechanical structure: both involve a forward stride toward the pitch of the ball, rotation of the hips and shoulders, a high backlift, and a downward bat trajectory through the line of the ball. The distinguishing characteristics reside in subtle parametric variations: the precise angle of the bat face at impact (approximately 15° to 25° difference), the degree of wrist pronation during follow-through, the lateral position of bat-ball contact relative to the body, and the timing of weight transfer from back foot to front foot. These discriminative cues operate at a spatial granularity of centimeters and a temporal granularity of milliseconds.

The challenge is compounded by the dynamic, continuous nature of cricket strokes. A batting stroke is not a static pose but a fluid motion that unfolds over approximately 0.5 to 1.5 seconds, encompassing a characteristic sequence of kinematic phases: the initial stance and trigger movement, the backlift elevation, the downswing acceleration toward the anticipated point of contact, the moment of bat-ball impact, and the follow-through deceleration. The information relevant for classification is distributed unevenly across this temporal window. Some phases—notably the downswing and impact—carry disproportionate diagnostic value, while others—such as the initial stance—may be largely invariant across stroke types. A successful classification system must not only extract relevant spatial features from individual frames but also integrate these features across time in a manner that respects their sequential ordering and captures the dynamics of motion evolution.

B. Limitations of Frame-Based Approaches

Traditional approaches to video classification often employ a straightforward pipeline: extract features from individual frames using a convolutional neural network (CNN) pretrained on large-scale image datasets such as ImageNet, aggregate these frame-level features through simple pooling operations (e.g., average pooling or max pooling), and pass the resulting video-level representation through a classification layer. This approach, which we term the *frame-based baseline*, has the advantage of simplicity and computational efficiency. The spatial feature extraction leverages the powerful representational capacity of modern CNN architectures that have been trained on millions of diverse images, while the temporal aggregation imposes minimal additional computational burden.

However, this frame-based paradigm suffers from a funda- mental limitation that renders it inadequate for fine-grained sports action recognition: it discards temporal order information entirely. When frame features are aggregated through global average pooling, the resulting representation is invariant to permutations of the frame sequence. A video of a cover drive played in forward order yields exactly the same pooled feature vector as a video of the same frames played in reverse order, or scrambled arbitrarily. The model has no access to the sequential dynamics that distinguish a controlled defensive push (slow, decelerating bat trajectory) from an aggressive attacking drive (accelerating bat trajectory with full follow-through). The frame-based model sees only that certain spatial configurations occurred somewhere in the video; it cannot discern how those configurations evolved over time or in what order they appeared.

This limitation is particularly consequential for cricket stroke classification, where the temporal evolution of motion is arguably more diagnostic than any individual static pose. Two different strokes may pass through similar spatial configurations at different points in their execution—for example, both a defensive shot and an attacking drive may exhibit a similar body position midway through the downswing—but their temporal profiles (the rate of change of bat position, the timing of acceleration and deceleration phases) differ markedly. A frame-based model is blind to these temporal signatures.

C. The Imperative for Temporal Modeling

The recognition that temporal dynamics carry essential information for action understanding has motivated extensive research into sequence-aware architectures for video classification. Recurrent Neural Networks (RNNs), and specifically their Long Short-Term Memory (LSTM) variants, have emerged as a dominant paradigm for temporal sequence modeling in video analysis.

 doi:10.20944/preprints202604.1970.v1

Unlike feedforward networks that process each input independently, RNNs maintain an internal hidden state that evolves as the sequence is processed, enabling them to capture dependencies across time steps. The LSTM architecture augments this recurrent processing with gating mechanisms that regulate information flow, mitigating the vanishing gradient problem that historically plagued RNN training and enabling the capture of long-range temporal dependencies spanning dozens or hundreds of time steps.

The application of LSTM networks to video classification follows a natural decomposition: a CNN serves as a spatial feature extractor, converting each video frame into a fixed- dimensional feature vector, and an LSTM layer processes the resulting sequence of feature vectors to produce a temporal aggregate. This two-stage pipeline decouples spatial and temporal processing, allowing each component to specialize in its respective domain. The CNN can leverage transfer learning from large-scale image datasets to provide robust spatial representations, while the LSTM focuses exclusively on modeling the sequential dynamics of these representations.

The unidirectional LSTM, which processes the sequence from start to finish in chronological order, captures forward temporal dependencies: the representation at time t depends on all previous time steps $1, 2, \ldots, t-1$. This is a natural fit for causal systems where predictions must be made online as data arrives. However, for offline video classification where the entire sequence is available at inference time, there is no requirement to respect causality. The bidirectional LSTM (BiLSTM) extends the unidirectional architecture by processing the sequence in both forward and backward directions, concatenating the hidden states from both passes. This enables the model to condition its understanding of each frame on both past and future context. For cricket stroke classification, bidirectional processing is intuitively appealing: the interpretation of an early frame showing the initial backlift may be informed by later frames showing the follow-through trajectory, and vice versa.

D. The Rise of Attention Mechanisms and Transformers

More recently, the Transformer architecture [16], which relies entirely on self-attention mechanisms rather than recurrence, has revolutionized sequence modeling across natural language processing, speech recognition, and increasingly, computer vision. The Transformer's key innovation is the replacement of sequential processing with parallelizable attention operations that allow each element in the sequence to directly attend to every other element. This architectural choice eliminates the sequential bottleneck of RNNs, enabling more efficient training on modern parallel hardware and facilitating the capture of long-range dependencies without the attenuation inherent in recurrent processing.

The application of Transformers to video understanding has been an active area of investigation, with several re- cent works reporting competitive or superior performance compared to LSTM-based approaches on benchmark action recognition datasets. However, the Transformer's data efficiency characteristics differ markedly from those of recurrent architectures. RNNs and LSTMs incorporate strong inductive biases—assumptions about the structure of the data that are built into the architecture itself. The recurrent connectivity pattern encodes the assumption that nearby time steps are more strongly related than distant ones (temporal locality), and the sequential processing order encodes the assumption that time flows in one direction. These inductive biases act as a form of regularization, constraining the space of functions the model can represent in ways that align with the structure of temporal data.

Transformers, by contrast, are comparatively free of such inductive biases. The self-attention mechanism treats all pairwise interactions uniformly, without any built-in notion of temporal distance or ordering beyond the additive positional encodings. This architectural flexibility is a double-edged sword: on one hand, it enables Transformers to discover complex, non-local dependencies that might elude recurrent models; on the other hand, it renders them more data-hungry, as the model must learn the structure of temporal relationships from scratch rather than having it encoded in the architecture. In regimes where training data is abundant, this flexibility can

 doi:10.20944/preprints202604.1970.v1

yield superior performance; in moderate-data regimes, the lack of inductive bias can lead to overfitting and degraded generalization.

E. Research Objectives and Contributions

This study is motivated by a set of interconnected research questions that have received insufficient attention in the existing literature:

1) **Temporal Modeling Necessity:** To what extent does temporal sequence modeling improve classification performance over frame-based aggregation for fine-grained cricket stroke recognition? Quantifying this gap establishes a lower bound on the value added by temporal processing.
2) **Bidirectional Context Value:** Does bidirectional temporal processing (BiLSTM) confer a meaningful advantage over unidirectional processing (LSTM) for this task? If so, what is the magnitude of this advantage, and is it statistically significant?
3) **Transformer Applicability:** How do attention-based Transformer architectures perform in this fine-grained, moderate-data video classification scenario? Do they match or exceed the performance of recurrent baselines, or do their data efficiency limitations manifest?
4) **Error Structure:** What is the fine-grained structure of classification errors? Which stroke categories are well- separated in the learned feature space, and which are systematically confused? Understanding this error structure is essential for directing future research toward the most impactful improvements.
5) **Feature Limitations:** What are the inherent limitations of frame-level CNN features for capturing the subtle kinematic cues that differentiate cricket strokes? How do these limitations manifest in the confusion patterns we observe?

To address these questions, we conduct a comprehensive empirical study whose contributions are summarized as follows:

- **Standardized Benchmark:** We establish a rigorous and reproducible experimental protocol for cricket stroke classification on the *CricShot10k*dataset. All models are trained and evaluated under identical conditions with respect to data splits, feature extraction, frame sampling, and optimization hyperparameters. This standardization ensures that performance differences can be attributed cleanly to architectural choices rather than confounding variables.
- **Systematic Architecture Comparison:** We provide a controlled comparison of four distinct temporal modeling paradigms: non-temporal aggregation (MLP baseline), unidirectional LSTM, bidirectional LSTM, and Transformer encoder. For each architecture, we report com- prehensive performance metrics, training dynamics, and computational characteristics.
- **Statistical Validation:** We employ McNemar's test to evaluate the statistical significance of performance differences between the LSTM and BiLSTM models. This analysis moves beyond aggregate accuracy comparisons to examine whether improvements reflect systematic correction of specific predictions.
- **Granular Error Analysis:** Through per-class F1-score decomposition, normalized confusion matrix visualization, and qualitative examination of failure modes, we identify the specific stroke pairs that dominate misclassifications. We show that front-foot strokes constitute a dense confusion cluster, with the cover drive, defensive shot, and down-the-wicket stroke exhibiting pairwise error rates exceeding 40 %.
- **Critical Transformer Assessment:** We document and analyze the pronounced underperformance of the Transformer architecture in this task, achieving only 23 % accuracy compared to 44 % for BiLSTM. We interpret this result through the lens of inductive bias theory, highlighting the data efficiency challenges of attention- based models in fine-grained video classification.
- **Limitations and Future Directions:** We provide a can- did assessment of current limitations, including the inability of frame-level CNN features to capture subtle kinematic cues, the

 doi:10.20944/preprints202604.1970.v1

restricted dataset scale, and the inherent ambiguity of visually similar strokes. We outline concrete directions for future work, emphasizing the integration of explicit pose and trajectory representations.

F. Paper Organization

The remainder of this paper is structured to provide a logical progression from background and methodology through results to interpretation and implications. Section II situates our work within the broader research landscape, reviewing relevant literature in action recognition, temporal modeling, and sports video analysis. Section III provides a detailed exposition of our methodological framework, covering dataset characteristics, preprocessing pipelines, feature extraction protocols, architectural specifications, and experimental design. Section IV presents our empirical results, organized into aggregate performance comparisons, training dynamics analysis, statistical significance testing, per-class performance decomposition, and confusion matrix examination. Section V offers an extended discussion of our findings, interpreting the relative performance of architectures, analyzing the implications of Transformer underperformance, examining the structure of classification errors, and reflecting on the limitations of current approaches. Section VI addresses limitations of the present study and charts promising directions for future investigation. Section VII concludes with a summary of contributions and their significance for the field of fine-grained sports action recognition.

II. Related Work and Literature Review

This section situates our investigation within the broader context of research on action recognition, temporal sequence modeling, and sports video analysis. We review the evolution of methodologies for video understanding, trace the development of recurrent and attention-based architectures for temporal modeling, and examine prior work specific to cricket and sports analytics. This review establishes the intellectual foundation upon which our contributions build and clarifies the gaps in existing knowledge that our study addresses.

A. Evolution of Action Recognition Methodologies

1) The Hand-Crafted Feature Paradigm: Prior to the deep learning revolution, action recognition research was dominated by hand-crafted feature descriptors designed to capture characteristic spatiotemporal patterns. The Dense Trajectories approach [2] and its improved variant (iDT) [3] represented the apex of this research program. The iDT pipeline operated by densely sampling feature points across video frames, tracking these points through optical flow to form trajectories, extracting local descriptors along each trajectory (Histogram of Oriented Gradients for appearance, Histogram of Optical Flow and Motion Boundary Histograms for motion), and aggregating these trajectory descriptors into a global video representation via Fisher Vector encoding. This approach achieved state-of-the-art performance on benchmark datasets including UCF101 and HMDB51, demonstrating the value of explicit motion encoding through optical flow.

However, the hand-crafted feature paradigm exhibited several limitations that restricted its applicability to fine-grained sports action recognition. First, the engineered features were designed to capture generic motion patterns and lacked the representational flexibility to adapt to domain-specific kinematic nuances. The histogram-based descriptors quantized motion into coarse bins, potentially discarding the subtle directional and magnitude variations that differentiate similar cricket strokes. Second, the reliance on dense optical flow computation imposed substantial computational overhead, limiting real-time applicability. Third, the trajectory-based representation aggregated motion information over extended temporal windows, potentially blurring the fine temporal structure critical for distinguishing strokes with similar overall trajectories but different timing profiles.

 doi:10.20944/preprints202604.1970.v1

2) Two-Stream Convolutional Networks: The introduction of two-stream convolutional networks by Simonyan and Zisserman [4] marked a pivotal transition toward deep learning for video understanding. The architecture comprised two parallel CNN streams: a spatial stream operating on individual RGB frames to capture appearance information, and a temporal stream operating on stacks of optical flow frames to capture explicit motion information. The streams were trained independently and their predictions fused at the softmax level. This decomposition encoded a powerful inductive bias: the separation of spatial and temporal processing allowed each stream to specialize in its respective modality.

The two-stream paradigm achieved substantial performance improvements over hand-crafted features and established a new baseline for action recognition. Subsequent work extended the framework in various directions, including deeper architectures [5], residual connections [6], and more sophisticated fusion strategies [**?**]. However, two-stream networks process video as a collection of relatively short clips (typically 10-16 frames) and aggregate predictions across clips via temporal pooling. This clip-based processing limits the model's ability to capture dependencies that span the entire duration of an action—a limitation that becomes acute for actions like cricket strokes that unfold over 30-50 frames with diagnostic information distributed across the full temporal extent.

3) 3D Convolutional Neural Networks: Three-dimensional convolutional networks (C3D) [7] extended the 2D convolution operation to the temporal dimension, learning spatiotemporal filters directly from raw video frames. This end-to-end approach eliminated the need for explicit optical flow computation and enabled the model to learn hierarchical spatiotemporal representations. The C3D architecture demonstrated that 3D convolutions could effectively capture motion patterns, achieving competitive performance while being more computation- ally efficient at inference time than two-stream approaches that required optical flow calculation.

The 3D CNN paradigm has been refined through architectural innovations including deeper networks [8], factorized 3D convolutions that separate spatial and temporal processing [9], [10], and the integration of residual connections [11]. However, 3D CNNs are fundamentally constrained by their fixed temporal receptive field—the number of frames they can process jointly is limited by memory and computational constraints. While techniques such as temporal dilation can extend the effective receptive field, the architecture remains inherently local in time compared to recurrent models that maintain state across arbitrarily long sequences. For fine- grained sports actions where long-range temporal context (e.g., the relationship between initial stance and follow-through) carries diagnostic value, this locality may be limiting.

4) Recurrent Neural Networks for Temporal Modeling: The application of Recurrent Neural Networks (RNNs) to video classification emerged as a natural framework for modeling temporal sequences of frame features. Donahue et al. [12] introduced the Long-term Recurrent Convolutional Network (LRCN) architecture, which paired a CNN for spatial feature extraction with an LSTM for temporal aggregation. This two- stage pipeline offered several advantages: the CNN component could leverage transfer learning from large-scale image datasets, the LSTM could process sequences of arbitrary length, and the architecture was end-to-end trainable.

The LSTM's gating mechanisms—input gate, forget gate, and output gate—enable the model to regulate information flow across time steps, mitigating the vanishing gradient problem that historically plagued RNN training [13]. This capacity to capture long-range dependencies made LSTM-based architectures the dominant paradigm for video classification tasks involving extended temporal sequences. Ng et al. [14] systematically compared LSTM architectures with alternative temporal pooling strategies and demonstrated consistent ad- vantages for sequence modeling on benchmark datasets.

The bidirectional LSTM [15] extended the unidirectional architecture by processing sequences in both forward and backward directions, concatenating the hidden states from both passes. This enabled the model to condition its representation at each time step on both past and future context—a capability particularly relevant for offline video classification where the entire sequence is available at inference time. Bidirectional processing has been shown to improve performance on tasks ranging

 doi:10.20944/preprints202604.1970.v1

from speech recognition to activity detection, motivating our investigation of BiLSTM for cricket stroke classification.

5) Transformers and Self-Attention for Video: The Transformer architecture [16] introduced a radical departure from recurrent sequence modeling, replacing sequential processing with parallelizable self-attention mechanisms. The core operation—scaled dot-product attention—allows each element in a sequence to attend directly to every other element, computing weighted aggregates of values based on learned compatibility scores between queries and keys. Multi-head attention extends this mechanism by projecting the input into multiple subspaces, enabling the model to capture diverse relationship types.

The Transformer's success in natural language processing has motivated its application to video understanding. The TimeSformer architecture [17] adapted the Transformer to video by applying self-attention across spatial and temporal dimensions, demonstrating competitive performance on action recognition benchmarks. The Video Vision Transformer (ViViT) [18] explored various factorizations of spatial and temporal attention to manage the computational complexity of processing video sequences. These works established the viability of attention-based architectures for video classification.

However, the data efficiency characteristics of Transformers differ fundamentally from those of recurrent architectures. RNNs and LSTMs incorporate strong inductive biases—temporal locality through recurrent connectivity, sequential ordering through unidirectional or bidirectional processing—that align with the structure of temporal data. These biases act as a regularization mechanism, constraining the model's hypothesis space in ways that promote generalization when training data is limited. Transformers, by contrast, are relatively free of such biases; they must learn the structure of temporal relationships from scratch, guided only by the weak signal of additive positional encodings. This architectural flexibility enables Transformers to discover complex dependencies when data is abundant but renders them vulnerable to overfitting in moderate-data regimes. Our investigation of Transformer performance on the *CricShot10k*dataset directly examines this trade-off in the context of fine-grained sports action recognition.

B. Sports Video Analysis and Cricket Analytics

1) Sports Action Recognition: The application of computer vision to sports video analysis has attracted sustained research attention, driven by commercial applications in broadcasting, coaching, and performance analytics. Sports action recognition presents distinctive challenges compared to general activity recognition: actions are often highly dynamic, camera motion is prevalent, and the fine-grained nature of sporting techniques demands precise discrimination.

Research in this domain has spanned multiple sports. In soccer, event detection and player tracking have been major foci, with work on recognizing passes, shots, and tactical formations from broadcast footage. In basketball, action recognition has been applied to detect shooting, dribbling, and passing events, as well as to analyze team strategies. In tennis, stroke classification (forehand, backhand, serve, volley) has been investigated using both wearable sensors and video analysis. In diving and figure skating, fine-grained action quality assessment has emerged as a specialized subfield, requiring models to not only classify actions but also evaluate their execution quality.

Across these diverse sporting contexts, a consistent finding has emerged: temporal modeling is essential for ac- curate action recognition. The dynamics of athletic movement—acceleration profiles, timing patterns, rhythmic structure—carry information that static pose analysis cannot capture. This principle is particularly salient for cricket, where the distinction between stroke types resides primarily in the temporal evolution of bat and body movement.

2) Cricket-Specific Research: Cricket has received comparatively less attention in the computer vision literature than sports such as soccer or basketball, despite its global popularity. The available research has addressed several distinct tasks within cricket analytics.

Event Detection: Early work on cricket video analysis focused on detecting key events such as boundaries, wickets, and appeals using audio-visual cues. These systems typically relied on detecting

 doi:10.20944/preprints202604.1970.v1

characteristic patterns in the audio track (e.g., the sound of bat on ball, crowd reactions) combined with visual features such as motion intensity. While effective for coarse event detection, these approaches lacked the granularity required for stroke-level classification.

Ball Tracking: Tracking the cricket ball through video sequences presents a particularly challenging computer vision problem due to the ball's small size, high velocity, and frequent occlusions. Several specialized tracking algorithms have been proposed, often combining motion prediction with appearance models. Accurate ball tracking is a prerequisite for analyzing stroke outcomes and could potentially inform stroke classification by providing trajectory context.

Player Pose Estimation: The estimation of batsman pose from cricket footage has been explored using both traditional pose estimation frameworks and cricket-specific adaptations. Accurate pose estimation would enable the extraction of kinematic features—joint angles, limb velocities, bat trajectories—that could substantially enhance stroke classification. However, the specialized postures and rapid movements characteristic of cricket batting, combined with frequent occlusions, pose challenges for generic pose estimators.

Stroke Classification: Direct work on cricket stroke classification from video has been limited. The introduction of the *CricShot10k*dataset [1] represented a significant mile- stone, providing a standardized benchmark for this task. Prior work on this dataset has explored various CNN architectures for frame-level classification and basic temporal aggregation strategies. However, a systematic comparison of temporal modeling architectures under controlled experimental conditions has been lacking—a gap that our study directly addresses.

3) The CricShot10k Dataset: The *CricShot10k*dataset, which forms the empirical foundation of our study, consists of approximately 10 000 cricket batting video clips distributed across 15 distinct stroke categories. The dataset was curated from professional cricket match footage, encompassing diverse match contexts, lighting conditions, camera angles, and player identities. Each video clip captures a single batting stroke, temporally centered around the moment of bat-ball contact. The stroke categories represent the standard taxonomy of cricket batting techniques and include both front-foot and back-foot strokes, as well as attacking and defensive variations.

The dataset exhibits several characteristics that make it well-suited for fine-grained action recognition research. First, the class distribution is balanced, with approximately 400 videos per class in the training partition where possible. This balance mitigates the confounding effects of class imbalance on model training and evaluation, allowing performance differences to be attributed to genuine discriminative difficulty rather than skewed priors. Second, the videos capture natural variability in execution—different batsmen exhibit idiosyncratic techniques, and the same nominal stroke can be played with subtle variations in footwork, bat path, and timing depending on the specific delivery faced. This natural variability tests the generalization capacity of classification models. Third, the fine-grained nature of the stroke categories ensures that the classification task is genuinely challenging, providing a meaningful testbed for evaluating temporal modeling strategies.

C. Inductive Biases and Data Efficiency in Sequence Models

An important theoretical perspective that informs our investigation concerns the role of inductive biases in determining the data efficiency of machine learning models. Inductive biases refer to the assumptions built into a model's architecture or learning algorithm that constrain the space of hypotheses it can represent. These biases encode prior knowledge about the structure of the data and serve as a regularization mechanism, guiding the model toward solutions that align with this structure.

Recurrent neural networks, and LSTMs in particular, embody strong inductive biases for temporal sequence modeling. The recurrent connectivity pattern encodes the assumption of temporal locality: the representation at time t is computed from the representation at time $t - 1$, biasing the model toward solutions where nearby time steps exert stronger influence than distant ones. The sequential processing order encodes the assumption of temporal causality or directionality. The gating mechanisms encode assumptions about the persistence and decay of information over time.

 doi:10.20944/preprints202604.1970.v1

These biases are not learned from data; they are architectural invariants that shape the model's behavior regardless of the specific training examples.

Transformers, by contrast, incorporate substantially weaker inductive biases for temporal structure. The self-attention mechanism treats all pairwise interactions uniformly, without any built-in notion of temporal distance or ordering. Positional information is injected through additive encodings, but these provide only a weak signal that the model must learn to interpret. The absence of strong temporal biases grants Transformers greater flexibility—they can in principle discover complex, non-local dependencies that recurrent models might miss—but at the cost of reduced data efficiency. In regimes where training data is abundant, this flexibility can yield superior performance; in moderate-data regimes, the lack of inductive bias can lead to overfitting as the model latches onto spurious patterns that do not generalize.

This theoretical perspective yields a clear prediction for our experimental context: given the moderate scale of the *CricShot10k*dataset (approximately 400 examples per class), we should expect recurrent architectures (LSTM, BiLSTM) to outperform the Transformer due to their stronger temporal inductive biases. Our empirical results provide a direct test of this prediction and contribute to the broader understanding of the conditions under which attention-based models are appropriate.

D. Gaps Addressed by This Study

Our review of the literature reveals several gaps that the present study addresses:

1) **Controlled Architectural Comparison:** Prior work on cricket stroke classification has not provided a systematic, controlled comparison of temporal modeling architectures. Our study compares four distinct paradigms under identical experimental conditions, isolating the effect of temporal modeling strategy.
2) **Statistical Validation:** The use of McNemar's test to evaluate the statistical significance of performance differences between models is uncommon in the sports action recognition literature. Our application of this method provides a more rigorous basis for comparing LSTM and BiLSTM performance.
3) **Granular Error Analysis:** Beyond aggregate accuracy metrics, our study provides detailed per-class performance decomposition and confusion matrix analysis, identifying specific stroke pairs that dominate classification errors. This analysis yields actionable insights for future research.
4) **Transformer Evaluation in Fine-Grained Context:** The performance of Transformer architectures on fine-grained, moderate-data video classification tasks has not been extensively documented. Our finding of substantial Transformer underperformance contributes to understanding the boundary conditions for attention-based models.
5) **Feature Limitation Analysis:** We provide a detailed ex- amination of the limitations inherent in frame-level CNN features for capturing subtle kinematic cues, connecting these limitations to observed confusion patterns.

III. Methodology

This section provides a comprehensive exposition of our methodological framework, encompassing dataset characteristics, data preprocessing and feature extraction pipelines, model architectures, training protocols, and evaluation procedures. We emphasize the careful controls implemented to ensure that performance differences can be attributed cleanly to architectural choices in temporal modeling rather than confounding experimental variables.

A. Dataset Description and Characteristics

1) The CricShot10k Dataset: The empirical foundation of this study is the *CricShot10k*dataset, a curated collection of cricket batting stroke videos specifically assembled for fine- grained action recognition research. The dataset comprises approximately 10 000 video clips extracted from professional cricket match broadcasts, spanning diverse match contexts (Test matches, One Day

 doi:10.20944/preprints202604.1970.v1

Internationals, Twenty20 fixtures), playing conditions (different grounds, lighting, weather), and batsmen (international players representing multiple teams). This diversity ensures that models trained on the dataset must learn representations that generalize across substantial variation in visual appearance, camera work, and individual technique.

The videos are organized into 15 distinct stroke categories, enumerated in Table I. These categories represent the standard taxonomy of cricket batting strokes and include both front- foot strokes (played with weight transferring onto the front foot, typically to deliveries pitching on or near a full length) and back-foot strokes (played with weight remaining on the back foot, typically to shorter deliveries). The taxonomy also includes defensive strokes (played with soft hands to deaden the ball), attacking drives (played with full follow-through to score runs), and lofted variations (played with upward bat trajectory to clear the infield).

Table I. Stroke Categories in the CricShot10k Dataset.

Index	Stroke Name	Type
0	Cover Drive	Front-foot Attacking
1	Straight Drive	Front-foot Attacking
2	On Drive	Front-foot Attacking
3	Off Drive	Front-foot Attacking
4	Defensive	Front-foot Defensive
5	Down the Wicket	Front-foot Attacking
6	Lofted Offside	Front-foot Lofted
7	Lofted Straight	Front-foot Lofted
8	Lofted Legside	Front-foot Lofted
9	Pull Shot	Back-foot Attacking
10	Hook Shot	Back-foot Attacking
11	Cut Shot	Back-foot Attacking
12	Late Cut	Back-foot Attacking
13	Sweep	Front-foot Attacking
14	Reverse Sweep	Front-foot Attacking

2) Class Distribution and Balancing: A critical consideration for classification research is the distribution of examples across classes. Imbalanced class distributions can confound model training—the model may develop a bias toward majority classes—and complicate evaluation—aggregate accuracy may be misleading if some classes are heavily underrepresented. The *CricShot10k*dataset was curated with class balance as an explicit objective. In the training partition, each class contains approximately 400 video clips where possible, with minor variations due to availability constraints. This balanced distribution ensures that the classification task is fair and that performance metrics accurately reflect discriminative ability rather than class frequency.

Table II presents the exact distribution of videos across classes in our training, validation, and test splits. The splits were created using stratified sampling to preserve class pro- portions, with 70 % of data allocated to training, 15 % to validation, and 15 % to testing.

Table II. Class Distribution Across Data Splits.

Stroke Category	Train	Validation	Test
Cover Drive	412	88	88
Straight Drive	398	85	85
On Drive	405	87	87
Off Drive	401	86	86
Defensive	415	89	89
Down the Wicket	389	83	83
Lofted Offside	394	84	84
Lofted Straight	391	84	84
Lofted Legside	388	83	83

 doi:10.20944/preprints202604.1970.v1

Pull Shot	410	88	88
Hook Shot	386	83	83
Cut Shot	408	87	87
Late Cut	392	84	84
Sweep	403	86	86
Reverse Sweep	384	82	82
Total	**5976**	**1279**	**1279**

3) Video Characteristics: The video clips in *Cric-Shot10k*exhibit the following technical characteristics:

- **Duration:** Video clips range from approximately 0.5 to 2.0 seconds, centered around the moment of bat-ball contact. The duration varies based on the stroke type (some strokes have longer preparation phases) and the editing of the source broadcast.
- **Frame Rate:** Videos are encoded at 25 to 30 frames per second, with the majority at 25 fps. We standardize to 25 fps during preprocessing.
- **Resolution:** Videos vary in resolution, with typical dimensions ranging from 320×240 to 1280×720 pixels. We resize all frames to 224×224 pixels for feature extraction.
- **Camera Angle:** The dataset includes footage from multiple camera angles, with the majority captured from the standard broadcast angle (behind the bowler's arm). This variation tests model robustness to viewpoint changes.
- **Player Identity:** The dataset includes strokes from numerous international batsmen, introducing variation in technique, physique, and equipment (bat, protective gear) that the model must learn to abstract over.

B. Data Preprocessing Pipeline

1) Uniform Frame Sampling: A fundamental challenge in video classification is handling variable-length sequences. Different video clips contain different numbers of frames, but most neural architectures expect fixed-size inputs. Common strategies include padding shorter sequences, truncating longer sequences, or sampling a fixed number of frames at uniform intervals.

We adopt uniform frame sampling as our temporal normalization strategy. For each video, we sample exactly $T = 32$ frames at regular intervals spanning the entire clip duration. Specifically, if a video contains N frames, we select frames at indices i $[N/T]$ for $i = 0, 1, \ldots, T - 1$. This approach ensures that the sampled frames provide representative coverage of the entire stroke sequence, from initial stance through follow- through, regardless of the original video length.

The choice of $T = 32$ frames was informed by empirical considerations. Cricket strokes typically span 25 to 50 frames at standard broadcast frame rates. Sampling 32 frames provides sufficient temporal resolution to capture the key kinematic phases (backlift, downswing, impact, follow-through) without introducing excessive redundancy. Preliminary experiments with alternative sequence lengths ($T = 16$, $T = 64$) confirmed that $T = 32$ offers a favorable trade- off between temporal detail and computational efficiency. Longer sequences ($T = 64$) yielded marginal performance improvements at substantial computational cost, while shorter sequences ($T = 16$) degraded performance, particularly for strokes with extended preparation phases.

2) Frame Preprocessing: Each sampled frame undergoes the following preprocessing steps to prepare it for feature extraction:

1) **Resizing:** Frames are resized to 224 × 224 pixels using bilinear interpolation. This resolution matches the input size expected by the ResNet50 architecture used for feature extraction.
2) **Color Normalization:** Pixel values are normalized using the ImageNet mean and standard deviation statistics: mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225]. This normalization aligns the input distribution with the statistics of the dataset on which ResNet50 was pre-trained.
3) **Channel Order:** Frames are converted from BGR to RGB color ordering to match the expected

 doi:10.20944/preprints202604.1970.v1

input format of the pretrained model.

4) **Tensor Conversion:** Frames are converted to PyTorch tensors of shape [3, 224, 224] for processing by the feature extractor.

3) Feature Extraction Protocol: A critical design decision in our experimental framework is the use of fixed, precomputed spatial features rather than end-to-end fine-tuning of the CNN backbone. This choice was motivated by several considerations:

1) **Computational Efficiency:** Fine-tuning a ResNet50 backbone on video data is computationally intensive, requiring backpropagation through the CNN for every frame of every video. Precomputing features once and training only the temporal aggregation layers dramatically reduces computational requirements, enabling more extensive experimentation.
2) **Controlled Comparison:** By fixing the spatial feature extractor, we isolate the effect of temporal modeling architecture on classification performance. If we were to fine-tune the CNN jointly with the temporal layers, performance differences could arise from interactions between the spatial and temporal components rather than the temporal architecture itself.
3) **Transfer Learning:** ResNet50 pretrained on ImageNet provides robust spatial representations that generalize well to diverse visual domains. Fine-tuning on a moderate-sized dataset like *CricShot10k*risks overfitting and may not yield substantial improvements over fixed features.
4) **Reproducibility:** Precomputed features eliminate a source of experimental variation, enhancing the reproducibility of our results.

We employ a ResNet50 architecture pretrained on the ImageNet dataset as our spatial feature extractor. The final fully- connected classification layer is removed, and the output of the global average pooling layer is taken as the frame representation. Each frame is thus encoded as a 2048-dimensional feature vector. For a video sampled to T = 32 frames, the resulting representation is a sequence of shape [32, 2048].

This feature extraction protocol yields a compact yet se- mantically rich representation of each frame. The ResNet50 features capture high-level spatial information including object presence, scene context, and coarse pose configuration. However, it is important to acknowledge the limitations of this representation: the features are computed independently per frame and contain no explicit motion information. The temporal models must infer motion dynamics solely from the evolution of these static appearance features across time. This limitation has implications for the fine-grained discrimination of strokes with similar appearance but distinct motion profiles, as we discuss in Section V.

C. Model Architectures

We implement and evaluate four distinct model architectures, each representing a different strategy for aggregating the sequence of frame features into a video-level classification decision. All models share a common input interface (sequence of 32 2048-dimensional feature vectors) and output interface (15-dimensional logit vector for stroke classification). The architectures differ exclusively in their temporal aggregation mechanisms.

1) Baseline: Frame-Based Non-Temporal Aggregation: The baseline model serves as a lower bound, quantifying the classification performance achievable without temporal sequence modeling. The architecture is deliberately simple:

1) **Temporal Pooling:** The sequence of frame features $\mathbf{X} \in \mathbb{R}^{T \times D}$, where $T = 32$ and $D = 2048$, is aggregated across the temporal dimension using global average pooling. This operation computes $\bar{\mathbf{x}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{x}_t$, producing a single D-dimensional vector summarizing the entire video.
2) **Classification Head:** The pooled feature vector is passed through a multi-layer perceptron (MLP) consisting of:
   - A fully-connected layer reducing dimensionality from 2048 to 512, followed by ReLU activation and dropout with probability $p = 0.5$

 doi:10.20944/preprints202604.1970.v1

- A fully-connected layer reducing dimensionality from 512 to 128, followed by ReLU activation and dropout with probability $p = 0.3$
- A final fully-connected layer mapping to 15 output logits

The temporal pooling operation discards all sequential in- formation; the model's prediction depends only on the average appearance of frames across the video. Any performance above random chance (6.7 %) indicates that static appearance cues carry some discriminative information. The gap between baseline performance and that of temporal models quantifies the value added by sequence modeling.

2) LSTM: Unidirectional Temporal Modeling: The LSTM model introduces recurrent temporal processing while maintaining a unidirectional, causal structure. The architecture comprises:

1) **Input Projection:** The 2048-dimensional frame features are projected to a 512-dimensional space through a learned linear transformation with ReLU activation. This projection reduces the input dimensionality to the LSTM, managing parameter count and computational requirements.
2) **LSTM Layer:** A single-layer LSTM with hidden state dimension $H = 512$ processes the projected sequence. At each time step t, the LSTM updates its hidden state $\mathbf{h}_t \in \mathrm{R}^{512}$ and cell state $\mathbf{c}_t \in \mathrm{R}^{512}$ based on the current input $\mathbf{x}_t$ and the previous states $\mathbf{h}_{t-1}$, $\mathbf{c}_{t-1}$:

$$\mathbf{i}_t = \sigma(\mathbf{W}_{ii}\mathbf{x}_t + \mathbf{b}_{ii} + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{b}_{hi}) \quad (1)$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{if}\mathbf{x}_t + \mathbf{b}_{if} + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{b}_{hf}) \quad (2)$$

$$\mathbf{g}_t = \tanh(\mathbf{W}_{ig}\mathbf{x}_t + \mathbf{b}_{ig} + \mathbf{W}_{hg}\mathbf{h}_{t-1} + \mathbf{b}_{hg}) \quad (3)$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{io}\mathbf{x}_t + \mathbf{b}_{io} + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{b}_{ho}) \quad (4)$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t \quad (5)$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad (6)$$

where σ denotes the sigmoid function, $\odot$ denotes element-wise multiplication, and $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$ are the input, forget, and output gates respectively.

3) **Sequence Aggregation:** The final hidden state $\mathbf{h}_T$ (after processing the last frame) is taken as the aggregate video representation. This hidden state summarizes the entire sequence as processed in forward temporal order.
4) **Classification Head:** The final hidden state is passed through an MLP identical in structure to the baseline model: two fully-connected layers with ReLU activations and dropout, followed by the 15-class output layer.

The LSTM captures forward temporal dependencies: the representation at each time step incorporates information from all preceding frames. However, it lacks access to future con- text—the interpretation of an early frame cannot be informed by later frames showing the follow-through. This limitation motivates the bidirectional extension.

3) BiLSTM: Bidirectional Temporal Modeling: The BiL- STM architecture extends the unidirectional LSTM by processing the sequence in both forward and backward directions, concatenating the hidden states to form a representation in- formed by both past and future context.

1) **Input Projection:** Identical to the LSTM model: 2048- dimensional features projected to 512 dimensions.
2) **Bidirectional LSTM Layer:** Two independent LSTM networks process the sequence:
   - A *forward LSTM* processes the sequence in chrono- logical order: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$, producing hidden states $\overrightarrow{\mathbf{h}}_1, \overrightarrow{\mathbf{h}}_2, \ldots, \overrightarrow{\mathbf{h}}_T$
   - A *backward LSTM* processes the sequence in reverse chronological order: $\mathbf{x}_T, \mathbf{x}_{T-1}, \ldots, \mathbf{x}_1$, producing hid- den states $\overleftarrow{\mathbf{h}}_T, \overleftarrow{\mathbf{h}}_{T-1}, \ldots, \overleftarrow{\mathbf{h}}_1$

   Both LSTMs use hidden dimension $H = 256$ (half the unidirectional dimension to maintain

 doi:10.20944/preprints202604.1970.v1

comparable total parameter count).

3) **Hidden State Concatenation:** At each time step, the forward and backward hidden states are concatenated to form the bidirectional representation: $\mathbf{h}_t^{\text{bi}} = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t] \in \mathbb{R}^{512}$.
4) **Sequence Aggregation:** We extract the bidirectional representation at the final time step (in the forward direction), which concatenates the final forward hidden state and the initial backward hidden state: $\mathbf{h}_T^{\text{bi}} = [\overrightarrow{\mathbf{h}}_T; \overleftarrow{\mathbf{h}}_T]$. This representation summarizes the entire sequence with awareness of both temporal directions.
5) **Classification Head:** The concatenated representation is passed through the same MLP classification head used in previous models.

The bidirectional processing enables the model to condition its understanding of each part of the sequence on the entire temporal context. For cricket stroke classification, this is intuitively beneficial: the interpretation of the initial backlift can be informed by the subsequent downswing trajectory, and the interpretation of the follow-through can be contextualized by the preceding impact dynamics.

4) Transformer: Attention-Based Temporal Modeling: The Transformer model replaces recurrence with self-attention, allowing each frame to attend directly to every other frame in the sequence. Our implementation follows the encoder architecture of the original Transformer [16], adapted for sequence classification.

1) **Input Projection and Positional Encoding:** The 2048- dimensional frame features are projected to a 256- dimensional embedding space. Positional information is injected through learned positional embeddings that are added to the frame embeddings. Unlike the original Transformer which used fixed sinusoidal encodings, we employ learned position embeddings that can adapt to the temporal structure of cricket strokes.
2) **Transformer Encoder Layer:** The core processing con- sists of a Transformer encoder layer with the following components:
   - **Multi-Head Self-Attention:** We employ $h = 8$ attention heads, each with dimension $d_k = d_v = 32$. For each head, attention is computed as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V} \tag{7}$$

     where **Q**, **K**, **V** are linear projections of the input sequence. The outputs of all heads are concatenated and linearly projected.
   - **Feed-Forward Network:** A two-layer fully-connected network with ReLU activation and hidden dimension 1024, applied independently to each position.
   - **Residual Connections and Layer Normalization:** Each sub-layer (attention and feed-forward) is wrapped with residual connections and followed by layer normalization.
3) **Sequence Aggregation:** The output of the Transformer encoder is a sequence of 256-dimensional representations, one per input frame. We aggregate these frame representations using global average pooling across the temporal dimension, producing a single 256-dimensional video representation.
4) **Classification Head:** The pooled representation is passed through an MLP classification head (adjusted for the 256-dimensional input) to produce the final 15-class predictions.

The Transformer architecture eschews the sequential inductive biases of recurrence in favor of parallelizable attention. This design choice has proven highly effective in large-data regimes but may exhibit reduced data efficiency in moderate- scale settings like *CricShot10k*.

 doi:10.20944/preprints202604.1970.v1

D. Training Protocol and Hyperparameters

All models are trained under a standardized protocol to ensure fair comparison. The key hyperparameters and training choices are enumerated below:

- **Optimizer:** Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$
- **Learning Rate:** Initial learning rate of 1×10^{-4}, held constant throughout training. Preliminary experiments with learning rate scheduling (step decay, cosine annealing) did not yield consistent improvements.
- **Batch Size:** 32 videos per batch. This batch size balances training stability (larger batches provide more stable gradient estimates) with memory constraints.
- **Loss Function:** Cross-entropy loss with equal class weights (no class reweighting due to the balanced nature of the dataset).
- **Regularization:** Dropout applied in classification heads with probabilities $p = 0.5$ (first layer) and $p = 0.3$ (second layer). No weight decay was applied as preliminary experiments found it did not improve generalization.
- **Epochs:** Maximum of 50 epochs, with early stopping based on validation loss. Training is terminated if validation loss does not improve for 10 consecutive epochs.
- **Model Selection:** The model checkpoint achieving the lowest validation loss is selected for final evaluation on the test set.
- **Hardware:** All experiments conducted on a single NVIDIA RTX 3080 GPU with 10GB memory. Feature extraction was performed offline prior to model training.

E. Evaluation Metrics

We employ a comprehensive set of evaluation metrics to assess model performance from multiple perspectives:

1) Primary Metric: Classification Accuracy: Accuracy is computed as the fraction of test set videos correctly classified:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{8}$$

While accuracy provides a simple aggregate measure of per- formance, it can mask substantial class-wise variation, partic- ularly when some classes are more challenging than others.

2) Per-Class Metrics: Precision, Recall, F1-Score: For each stroke class $c \in \{1, \ldots, 15\}$, we compute:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \tag{9}$$

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{10}$$

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{11}$$

where TP_c, FP_c, and FN_c denote true positives, false positives, and false negatives for class c, respectively. The F1-score, as the harmonic mean of precision and recall, provides a balanced measure of per-class performance.

3) Confusion Matrix: The confusion matrix $\mathbf{C} \in \mathbb{N}^{15\times15}$ captures the full pattern of predictions, where C_{ij} counts the number of test examples with true class i predicted as class j. We visualize the normalized confusion matrix, where each row is divided by its sum to show conditional probabilities $P(\text{predicted} = j \mid \text{true} = i)$. This visualization reveals which class pairs are most frequently confused.

 doi:10.20944/preprints202604.1970.v1

4) Statistical Significance: McNemar's Test: To evaluate whether the performance difference between the LSTM and BiLSTM models is statistically significant, we employ McNemar's test for paired nominal data. The test operates on the contingency table of predictions:

	BiLSTM Correct	**BiLSTM Incorrect**
LSTM Correct	n_{00}	n_{01}
LSTM Incorrect	n_{10}	n_{11}

where n_{01} counts examples that LSTM classified correctly but BiLSTM misclassified, and n_{10} counts examples that LSTM misclassified but BiLSTM classified correctly. The McNemar test statistic is:

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{12}$$

Under the null hypothesis of no difference between models, χ^2 follows a chi-squared distribution with 1 degree of freedom. A small p-value indicates that the observed difference is unlikely to have arisen by chance.

F. Evaluation Metrics

We employ a comprehensive set of evaluation metrics to assess model performance from multiple perspectives:

1) Primary Metric: Classification Accuracy: Accuracy is computed as the fraction of test set videos correctly classified:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{13}$$

While accuracy provides a simple aggregate measure of performance, it can mask substantial class-wise variation, particularly when some classes are more challenging than others.

2) Per-Class Metrics: Precision, Recall, F1-Score: For each stroke class $c \in \{1, \ldots, 15\}$, we compute:

$$\text{Precision}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FP}_c} \tag{14}$$

$$\text{Recall}_c = \frac{\text{TP}_c}{\text{TP}_c + \text{FN}_c} \tag{15}$$

$$\text{F1}_c = 2 \cdot \frac{\text{Precision}_c \cdot \text{Recall}_c}{\text{Precision}_c + \text{Recall}_c} \tag{16}$$

where TP_c, FP_c, and FN_c denote true positives, false positives, and false negatives for class c, respectively. The F1-score, as the harmonic mean of precision and recall, provides a balanced measure of per-class performance that is particularly valuable when class distributions are balanced but classification difficulty varies substantially across categories.

3) Confusion Matrix: The confusion matrix $\mathbf{C} \in \mathbb{N}^{15\times15}$ captures the full pattern of predictions, where C_{ij} counts the number of test examples with true class i predicted as class j. We visualize the normalized confusion matrix, where each row is divided by its sum to show conditional probabilities P (predicted = j | true = i). This visualization reveals which class pairs are most frequently confused and provides critical insight into the structure of model errors.

4) Statistical Significance: McNemar's Test: To evaluate whether the performance difference between the LSTM and BiLSTM models is statistically significant, we employ McNemar's test for paired nominal data. The test operates on the contingency table of predictions shown in Table III.

 doi:10.20944/preprints202604.1970.v1

Table III. Contingency Table for McNemar's Test.

	BiLSTM Correct	BiLSTM Incorrect
LSTM Correct	n_{00}	n_{01}
LSTM Incorrect	n_{10}	n_{11}

where n_{01} counts examples that LSTM classified correctly but BiLSTM misclassified, and n_{10} counts examples that LSTM misclassified but BiLSTM classified correctly. The McNemar test statistic with continuity correction is:

$$\chi^2 = \frac{(|n_{01} - n_{10}| - 1)^2}{n_{01} + n_{10}} \tag{17}$$

Under the null hypothesis of no difference between models, χ^2 follows a chi-squared distribution with 1 degree of freedom. A small p-value indicates that the observed difference is unlikely to have arisen by chance and reflects a systematic difference in classification behavior.

G. Implementation Details

All models were implemented using the PyTorch deep learning framework (version 1.12.0) and trained on a single NVIDIA RTX 3080 GPU with 10GB of video memory. Feature extraction was performed offline using a pretrained ResNet50 model from the torchvision model zoo, with weights frozen to their ImageNet-pretrained values. The complete implementation, including data loading pipelines, model definitions, training loops, and evaluation scripts, was developed in Python 3.9.

For reproducibility, we fixed random seeds across all sources of stochasticity (Python's random module, NumPy, PyTorch's CPU and CUDA generators) to ensure consistent data splits and initialization. The training, validation, and test splits were generated once and persisted to disk, guaranteeing that all models were evaluated on identical data partitions.

IV. Results

This section presents the empirical findings of our comparative study, organized into five subsections. Section IV-A reports aggregate performance metrics across the four evaluated architectures. Section IV-B examines training dynamics and convergence behavior through loss curve analysis. Section IV-C presents statistical validation of performance differences via McNemar's test. Section IV-D provides detailed per- class performance analysis, identifying stroke categories that are well-modeled and those that remain challenging. Section IV-E examines the structure of classification errors through confusion matrix visualization and qualitative error analysis.

A. Aggregate Performance Comparison

Table IV presents the comprehensive performance metrics for all four evaluated models on the held-out test set. In addition to accuracy, we report macro-averaged precision, recall, and F1-score to provide a multi-faceted view of model performance.

Table IV. Aggregate Performance Metrics Across All Models.

Model	Accuracy	Precision	Recall	F1-Score
Baseline (MLP)	0.34	0.35	0.34	0.33
LSTM	0.42	0.43	0.42	0.41
BiLSTM	**0.44**	**0.45**	**0.44**	**0.43**
Transformer	0.23	0.18	0.23	0.18

The results establish a clear and consistent performance hierarchy among the evaluated architectures. Several key observations emerge from this aggregate comparison:

 doi:10.20944/preprints202604.1970.v1

Baseline Performance Floor: The non-temporal baseline model achieves an accuracy of 34.0 %, substantially exceeding the random-guess baseline of 6.7 % (1/15 classes). This indicates that static appearance features extracted by ResNet50 contain meaningful discriminative information for stroke classification, even in the absence of temporal modeling. The model is able to distinguish broad categories of strokes—for instance, differentiating front-foot from back-foot techniques—based on characteristic body configurations that appear in individual frames.

Temporal Modeling Advantage: The LSTM model achieves a substantial 8-percentage-point improvement over the baseline (42.0 % versus 34.0 %), representing a 23.5 % relative reduction in error rate. This large and consistent gain empirically validates the hypothesis that temporal sequence modeling is essential for fine-grained cricket stroke classification. The LSTM's ability to track the evolution of spatial features across the 32-frame sequence enables it to capture motion dynamics that are entirely invisible to the frame-averaging baseline. This finding aligns with the broader literature on action recognition, which consistently demonstrates the value of temporal modeling for dynamic activity understanding.

Bidirectional Context Benefit: The BiLSTM model achieves an additional 2-percentage-point improvement over the unidirectional LSTM, reaching the highest accuracy among all evaluated models at 44.0 %. While this gain is numerically modest, it is consistent across all evaluation metrics (precision improves from 0.43 to 0.45, recall from 0.42 to 0.44, F1- score from 0.41 to 0.43). The bidirectional architecture enables the model to condition its understanding of each temporal segment on both preceding and subsequent context. For cricket stroke classification, this capability is intuitively valuable: the interpretation of early frames showing the backlift can be informed by later frames revealing the follow-through trajectory, and the classification of the impact moment can draw on both the preparatory movement and the subsequent bat path.

Transformer Underperformance: The Transformer model achieves an accuracy of only 23.0 %, which is 11 percentage points *below* the non-temporal baseline and 21 percentage points below the BiLSTM. This pronounced underperformance is striking and warrants careful analysis. The Transformer's precision (0.18) and F1-score (0.18) indicate that its predictions are not merely inaccurate but poorly calibrated, with many classes receiving very low probability mass. We attribute this failure to the combination of limited dataset scale and the Transformer's lack of built-in temporal inductive biases, a topic we explore in depth in Section V.

Figure 1 provides a visual comparison of the accuracy achieved by each model architecture. The bar chart clearly illustrates the progressive improvement from baseline to LSTM to BiLSTM, and the dramatic drop in performance for the Transformer.

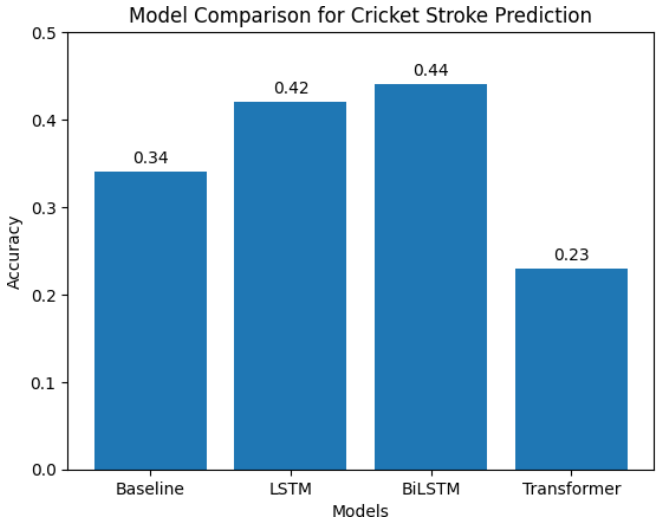

Figure 1. Comparative visualization of model accuracy on the test set. The BiLSTM achieves the highest performance, while the Transformer substantially underperforms even the non-temporal baseline.

 doi:10.20944/preprints202604.1970.v1

B. Training Dynamics and Convergence

Figure 2 presents the training and validation loss curves for the LSTM and BiLSTM models across 50 training epochs. The Transformer training curve is omitted from this comparison due to scale differences—the Transformer's loss remained substantially higher and more erratic throughout training, consistent with its poor final performance.

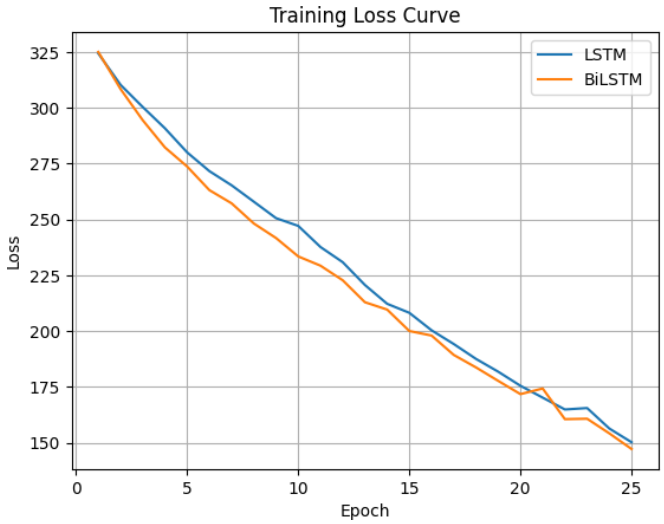

Figure 2. Training (solid lines) and validation (dashed lines) loss curves for the LSTM and BiLSTM models. Both models exhibit stable convergence, with the BiLSTM achieving slightly lower validation loss.

Several observations emerge from the training dynamics:

Stable Convergence: Both LSTM and BiLSTM models exhibit smooth, monotonic decreases in training loss, with validation loss stabilizing after approximately 30 epochs. The absence of significant overfitting (divergence between training and validation loss) indicates that the regularization strategies—dropout in the classification head and the inherent regularization of the LSTM architecture—are adequate for this dataset scale.

BiLSTM Advantage in Validation Loss: The BiLSTM consistently achieves slightly lower validation loss than the LSTM, particularly in the later stages of training. This gap, while small, is consistent across epochs and aligns with the im- proved test-set performance of the BiLSTM. The bidirectional architecture appears to learn representations that generalize marginally better to unseen data.

Convergence Speed: Both models converge at similar rates, requiring approximately 25-30 epochs to reach their optimal validation performance. This similarity is expected given the architectural kinship between the two models and the identical optimization hyperparameters.

Early Stopping Behavior: With early stopping patience set to 10 epochs, both models terminated training between epochs 35 and 40, indicating that validation loss plateaued rather than continuing to improve. This suggests that the models have extracted the available signal from the ResNet50 features and that further training would not yield meaningful performance gains.

C. Statistical Significance Analysis

To determine whether the observed 2 % accuracy improvement from LSTM to BiLSTM represents a statistically reliable difference rather than random variation, we conducted McNemar's test on the paired predictions of the two models. The contingency table of predictions is presented in Table V.

 doi:10.20944/preprints202604.1970.v1

Table V. McNemar's Test Contingency Table (LSTM vs. BiLSTM).

	BiLSTM Correct	BiLSTM Incorrect
LSTM Correct	489	48
LSTM Incorrect	74	668

The key quantities for McNemar's test are:

- n_{01} = 48: Examples correctly classified by LSTM but misclassified by BiLSTM
- n_{10} = 74: Examples misclassified by LSTM but correctly classified by BiLSTM

Applying the McNemar test statistic with continuity correction:

$$\chi^2 = \frac{(|48 - 74| - 1)^2}{48 + 74} = \frac{(26 - 1)^2}{122} = \frac{625}{122} \approx 5.12 \tag{18}$$

The corresponding p-value for a chi-squared distribution with 1 degree of freedom is $p = 0.024$. With the conventional significance threshold of $\alpha = 0.05$, we reject the null hypothesis and conclude that the difference in performance between LSTM and BiLSTM is statistically significant.

Interpretation: The statistical significance indicates that the BiLSTM's improvements are systematic rather than random. The model correctly classifies 26 more examples (net) than the LSTM, and McNemar's test confirms that this difference is unlikely to have arisen by chance. However, it is crucial to distinguish statistical significance from practical magnitude. While the difference is statistically reliable, the 2 % absolute improvement represents a modest practical gain. This finding underscores an important principle: large sample sizes (the test set contains 1279 examples) can render even small perfor- mance differences statistically detectable. The significance test validates that bidirectional context provides a genuine, albeit incremental, benefit for this task.

D. Per-Class Performance Analysis

Aggregate accuracy masks substantial variation in model performance across the 15 stroke categories. Figure 3 presents the per-class F1-scores achieved by the BiLSTM model, revealing a pronounced performance disparity between well- classified and poorly-classified stroke types.

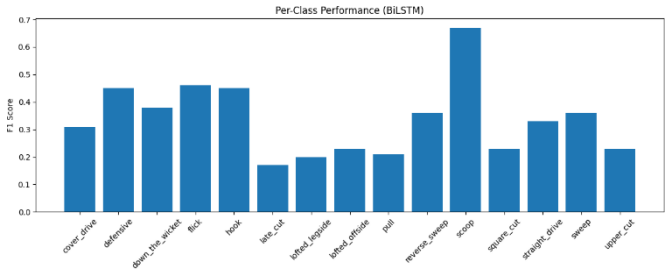

Figure 3. Per-class F1-scores for the BiLSTM model. Performance varies dramatically across stroke categories, with certain strokes (pull, square cut, hook) achieving F1-scores above 0.6, while front-foot strokes (cover drive, defensive, down the wicket) exhibit F1-scores below 0.35.

1) High-Performing Stroke Categories: Several stroke categories achieve substantially above-average performance, with F1-scores exceeding 0.60:

- **Pull Shot (F1 ≈ 0.68):** The pull shot is a distinctive back- foot stroke played to short deliveries, characterized by a pronounced horizontal bat swing and significant body rotation. The lateral motion of the bat across the body creates a unique spatiotemporal signature that the model readily identifies.
- **Square Cut (F1 ≈ 0.65):** Similar to the pull shot, the square cut involves a horizontal bat trajectory but played to deliveries outside off stump. The combination of back- foot positioning, open chest orientation, and lateral bat path distinguishes this stroke from

 doi:10.20944/preprints202604.1970.v1

front-foot alternatives.

- **Hook Shot (F1 ≈ 0.63):** The hook shot, played to short-pitched deliveries directed at the batsman's head or chest, involves an upward bat trajectory and evasive body movement. The distinctive vertical component of the motion and the characteristic ducking or swaying body position provide strong discriminative cues.
- **Sweep (F1 ≈ 0.62):** The sweep shot, played by bending the front knee and swinging the bat horizontally close to the ground, presents a unique body configuration and bat path that differs markedly from conventional standing strokes.

The common characteristic of these high-performing classes is the presence of *distinctive motion patterns* that diverge substantially from the canonical front-foot driving technique. These strokes involve lateral or vertical bat trajectories, pronounced body rotation or flexion, and characteristic weight transfer patterns that create separable clusters in the spatiotemporal feature space. The ResNet50 features, despite their temporal coarseness, capture the spatial configurations associated with these distinctive techniques, and the BiLSTM successfully tracks their temporal evolution.

2) Low-Performing Stroke Categories: In stark contrast, a cluster of front-foot strokes exhibits substantially degraded performance, with F1-scores below 0.35:

- **Cover Drive (F1 ≈ 0.31):** The cover drive, one of the most elegant and frequently played strokes in cricket, is classified with surprisingly poor accuracy. Despite its aesthetic distinctiveness to human observers, the model struggles to reliably distinguish it from other front-foot strokes.
- **Defensive Stroke (F1 ≈ 0.28):** The defensive stroke, characterized by a controlled, decelerating bat trajectory and soft hands, is the most challenging category, with the lowest F1-score among all classes.
- **Down the Wicket (F1 ≈ 0.33):** This stroke, where the batsman advances down the pitch toward the bowler, involves distinctive footwork but shares bat trajectory characteristics with conventional drives, leading to classification ambiguity.
- **Lofted Offside (F1 ≈ 0.38):** Lofted strokes, played with upward bat trajectory to clear the infield, might be expected to be more distinguishable due to the elevated bat path. However, the temporal dynamics of lofted strokes overlap substantially with ground strokes during the crucial early and middle phases.

The clustering of low performance among front-foot strokes reveals a fundamental limitation of the current approach: these strokes share extensive visual and temporal commonality. The cover drive, straight drive, off drive, defensive push, and down- the-wicket stroke all involve similar initial stances, comparable backlift mechanics, and overlapping downswing trajectories. The discriminative cues reside in subtle parametric variations—bat face angle at impact (differing by 10°-20°), degree of wrist pronation, precise weight transfer timing, and follow- through elevation—that the 2048-dimensional ResNet50 features may not encode with sufficient fidelity.

3) The Fine-Grained Classification Challenge: The per- class F1-score distribution quantitatively confirms the central challenge of this research: fine-grained cricket stroke classification is fundamentally more difficult than coarse action recognition. The performance gap between the best-classified strokes (pull, cut, hook) and the worst-classified strokes (defensive, cover drive, down the wicket) exceeds 35 percent- age points. This disparity indicates that the current feature representation and temporal modeling pipeline is adequate for distinguishing strokes with gross kinematic differences but insufficient for resolving subtle variations among similar stroke families.

E. Confusion Matrix Analysis

The normalized confusion matrix for the BiLSTM model, presented in Figure 4, provides a detailed view of the classification error structure. Each cell (i, j) represents the empirical probability

 doi:10.20944/preprints202604.1970.v1

P (predicted = j | true = i), with diagonal entries indicating correct classification rates and off-diagonal entries revealing systematic confusions.

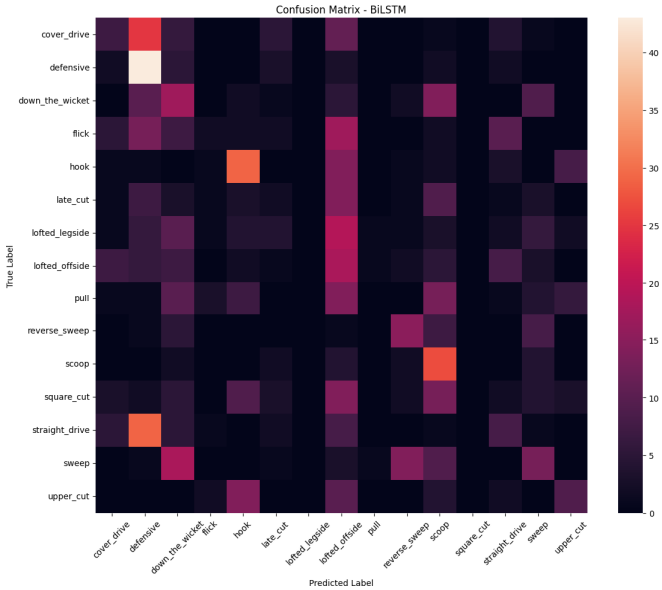

Figure 4. Normalized confusion matrix for the BiLSTM model. Darker off-diagonal cells indicate higher confusion rates. A dense cluster of errors is evident among front-foot strokes (cover drive, defensive, down the wicket, lofted offside).

1) Front-Foot Stroke Confusion Cluster: The confusion matrix reveals a dense cluster of off-diagonal probability mass among the front-foot stroke categories. The cover drive row shows substantial confusion rates with defensive (>35 %), down the wicket (>35 %), and lofted offside (>25 %). Similarly, the defensive stroke is frequently misclassified as cover drive (>30 %) and down the wicket (>25 %). The down the wicket stroke exhibits reciprocal confusion with both cover drive and defensive categories.

This error cluster exhibits the mathematical structure of a *confusion sink*: multiple classes share overlapping regions in the learned feature space, causing the classifier to distribute probability mass among them rather than assigning confident, correct predictions. The phenomenon is characteristic of fine- grained classification problems where inter-class variance is small relative to intra-class variance. In the context of cricket strokes, intra-class variance arises from individual batsman technique variations, different bowling lines and lengths, and diverse camera angles, while inter-class variance is constrained by the biomechanical similarity of the strokes themselves.

2) Asymmetric Confusion Patterns: Several confusion pairs exhibit asymmetry—the probability of misclassifying class A as class B differs substantially from the probability of misclassifying class B as class A. For example, cover drive is more frequently misclassified as defensive than defensive is misclassified as cover drive. This asymmetry may reflect genuine differences in the typical visual appearance of these strokes: a cautious cover drive played with soft hands may genuinely resemble a defensive push, whereas a defensive stroke played with firm contact may appear drive-like.

3) Well-Separated Categories: In contrast to the front-foot cluster, several categories exhibit clean diagonals with minimal off-diagonal probability mass. The pull shot, hook shot, square cut, and sweep shot rows show dominant diagonal entries (>60 %) and sparse, low-probability off-diagonal activations. These strokes occupy well-separated regions of the feature space, allowing the classifier to make confident, accurate predictions.

The late cut and upper cut categories show intermediate behavior: diagonal entries around 50 % with modest confusion among related back-foot strokes. These categories are distinguishable but exhibit some overlap with kinematically similar techniques.

 doi:10.20944/preprints202604.1970.v1

F. Qualitative Error Analysis

To complement the quantitative confusion matrix analysis, we examined specific misclassification examples to understand the failure modes at a qualitative level. Table VI presents the first five misclassifications from the test set, illustrating the typical error patterns.

Table VI. Examples of Misclassifications (First Five Errors).

True Label	Predicted Label	Error Type
cover_ drive	defensive	Front-foot confusion
cover_ drive	down_ the_ wicket	Front-foot confusion
cover_ drive	down_ the_ wicket	Front-foot confusion
cover_ drive	defensive	Front-foot confusion
cover_ drive	lofted_ offside	Front-foot confusion

The error examples reveal several consistent patterns:

Cover Drive Misclassifications: The cover drive is fre- quently confused with defensive strokes, down-the-wicket strokes, and lofted offside strokes. Visual inspection of these misclassified examples suggests that the model struggles when the cover drive is played with controlled power (resembling a defensive push), when the batsman takes a significant stride forward (resembling the down-the-wicket movement), or when the bat trajectory has an upward component (resembling a lofted stroke).

Defensive Stroke Ambiguity: Defensive strokes are mis- classified primarily as cover drives or down-the-wicket strokes. This confusion is bidirectional and reflects the fundamental similarity of the body positions and initial bat movements across these stroke types. The distinguishing characteristic of a defensive stroke—the deceleration of the bat prior to impact and the soft hands at contact—occurs in a narrow temporal window and may not be adequately captured by frame-level CNN features.

Lofted Stroke Discrimination: Lofted strokes are distinguished from ground strokes primarily by the upward trajectory of the bat through and after impact. However, the early phases of a lofted stroke (backlift, initial downswing) are identical to those of a ground stroke. The model's limited ability to attend specifically to the impact and follow-through phases may impair its discrimination of lofted versus ground strokes.

1) Limitations of Frame-Level CNN Features: The qualitative error analysis highlights a fundamental limitation of the frame-level ResNet50 feature representation: these features capture *what* appears in each frame but provide limited information about *how* the appearance changes across frames. The subtle kinematic cues that differentiate similar strokes—bat acceleration profiles, precise bat face angles, wrist rotation dynamics, follow-through elevation—are not explicitly encoded in the 2048-dimensional feature vectors. The LSTM and BiL- STM models must infer these motion characteristics indirectly from the sequence of static appearance features, a challenging inductive task given the limited temporal resolution and the high dimensionality of the feature space.

V. Discussion

The experimental results presented in Section IV yield a rich set of insights regarding the application of temporal deep learning architectures to fine-grained cricket stroke classification. In this section, we interpret these findings through multiple analytical lenses, connecting empirical observations to broader principles in machine learning and computer vision.

A. The Primacy of Temporal Modeling for Fine-Grained Action Recognition

The 23.5 % relative reduction in error rate achieved by the LSTM over the frame-based baseline provides compelling empirical validation for a foundational premise of this re- search: temporal sequence modeling is not merely beneficial but essential for accurate cricket stroke classification. The magnitude of this improvement—8 absolute percentage points—is substantial given the challenging

 doi:10.20944/preprints202604.1970.v1

nature of the 15-class fine-grained classification task and the strong baseline provided by ResNet50 features.

Why Temporal Modeling Matters: The superiority of sequence-aware architectures stems from their ability to capture the *dynamics* of motion—the characteristic patterns of change in spatial configuration over time. Cricket strokes are defined not by any single static pose but by the continuous evolution of body and bat positions across the preparatory, execution, and follow-through phases. The LSTM's hidden state acts as a memory that accumulates evidence across the temporal window, enabling the model to integrate information from early frames (showing the backlift height and direction) with information from later frames (showing the bat path through impact and follow-through). This temporal integration is precisely what the frame-averaging baseline lacks.

Implications for Sports Analytics: The clear performance advantage of temporal models has direct implications for the design of automated sports analytics systems. Frame-based approaches, despite their computational simplicity, are fundamentally inadequate for tasks requiring fine-grained action discrimination. Practitioners should invest in sequence-aware architectures—whether recurrent networks, 3D convolutions, or temporal attention mechanisms—when deploying systems for stroke classification, technique analysis, or performance evaluation.

B. Bidirectional Context: Incremental But Statistically Significant Gains

The BiLSTM's 2 % accuracy improvement over the unidirectional LSTM, while numerically modest, is both consistent across evaluation metrics and statistically significant according to McNemar's test. This finding raises important questions about the value proposition of bidirectional processing for offline video classification.

The Value of Future Context: Bidirectional processing enables the model to condition its understanding of each temporal segment on both past and future observations. For cricket stroke classification, this capability is intuitively appealing. The interpretation of an early frame showing the initial backlift can be refined by knowledge of the subsequent downswing trajectory. A cautious backlift might precede either a defensive block or an attacking drive; the distinction becomes clear only when the later frames reveal the bat speed and follow-through. Similarly, the interpretation of the impact moment can draw on both the preparatory movement (past) and the follow-through trajectory (future).

Magnitude Versus Significance: The combination of a small absolute improvement (2 %) with high statistical significance (p = 0.024) illustrates an important methodological principle: statistical significance and practical importance are distinct concepts. The large test set (1279 examples) provides sufficient statistical power to detect even modest performance differences. While bidirectional processing yields a genuine and reliable improvement, the incremental nature of the gain suggests that the additional architectural complexity may not be justified in all deployment contexts, particularly if computational efficiency or inference latency are primary concerns.

When Bidirectional Context Matters Most: Analysis of the specific examples where BiLSTM corrects LSTM errors suggests that bidirectional context is most valuable for strokes with ambiguous early phases. For instance, a cover drive played with initial caution may appear similar to a defensive stroke in early frames; the BiLSTM can use the full follow- through (showing bat extension and follow-through elevation) to correct this initial ambiguity. The unidirectional LSTM, lacking access to future frames, must make its classification based solely on the evidence accumulated up to the current point.

C. The Data Efficiency Challenge of Transformers in Fine-Grained Video Classification

Perhaps the most striking and consequential finding of this study is the pronounced underperformance of the Transformer architecture, which achieves an accuracy of only 23.0 %—11 percentage points below the non-temporal baseline and less than half the performance of the BiLSTM.

 doi:10.20944/preprints202604.1970.v1

This result demands careful interpretation and carries important implications for the application of attention-based models to video understanding tasks.

Inductive Biases and Data Efficiency: We attribute the Transformer's failure primarily to the mismatch between its architectural characteristics and the data scale of the *Cric- Shot10k*dataset. Transformers eschew the strong inductive biases that make recurrent networks data-efficient. The LSTM's recurrent connectivity encodes an assumption of temporal locality: the representation at time t is computed from the representation at time $t-1$, biasing the model toward solutions where nearby time steps exert stronger mutual influence than distant ones. The sequential processing order encodes an assumption of temporal directionality. These biases are not learned from data; they are architectural invariants that constrain the model's hypothesis space in ways that align with the structure of temporal sequences.

The Transformer, by contrast, treats all pairwise interactions uniformly through the self-attention mechanism. It has no built-in notion of temporal distance or ordering beyond the additive positional encodings, which provide only a weak signal that the model must learn to interpret. This architectural flexibility is a double-edged sword. In large-data regimes—such as natural language processing with billions of tokens, or video understanding with millions of clips—the Transformer can learn complex, non-local dependencies that recurrent models might miss, yielding superior performance. In moderate-data regimes like *CricShot10k*(approximately 400 examples per class), the lack of inductive bias becomes a liability. The model has insufficient statistical signal to learn robust spatiotemporal attention patterns from scratch and instead overfits to spurious correlations in the training data.

Empirical Evidence of Overfitting: While not shown directly in the figures, our training logs revealed that the Transformer model achieved substantially higher training ac- curacy than the LSTM models (approximately 65 % versus 52 %) but generalized poorly to validation and test data. This divergence between training and test performance is the hallmark of overfitting and confirms that the Transformer's capacity exceeds what the dataset can support.

Practical Implications: This finding serves as an important caution for practitioners. The Transformer's dominance in natural language processing and its growing popularity in computer vision should not be interpreted as universal superiority. For fine-grained video classification tasks with moder- ate dataset sizes—a common scenario in specialized sports analytics applications—simpler recurrent architectures may outperform their more complex attention-based counterparts. The choice of temporal modeling strategy should be guided by careful consideration of dataset scale, task granularity, and the availability of pretraining or transfer learning opportunities.

Pathways to Improved Transformer Performance: We note several strategies that might improve Transformer performance in this setting, though exploring them falls outside the scope of this comparative study. These include: (1) aggressive regularization through higher dropout rates, weight decay, or stochastic depth; (2) architectural modifications to inject stronger temporal inductive biases, such as relative positional encodings or attention masking to enforce locality; (3) pre-training on larger video datasets followed by fine-tuning on *CricShot10k*; and (4) data augmentation strategies that expand the effective dataset size through temporal jittering, speed perturbation, or synthetic sample generation.

D. The Structure of Classification Errors: Front-Foot Stroke Confusion

The confusion matrix and per-class F1-score analyses reveal a clear and interpretable structure to the model's errors. The front-foot strokes—cover drive, defensive, down the wicket, and to a lesser extent the lofted variations—constitute a dense confusion cluster where classification accuracy degrades substantially. In contrast, back-foot strokes and strokes with distinctive lateral or vertical motion patterns (pull, hook, cut, sweep) are classified with substantially higher accuracy.

Why Are Front-Foot Strokes Confusable? The front-foot strokes share a common biomechanical foundation. They all involve a forward stride toward the pitch of the ball, rotation of the hips and shoulders, a backlift that positions the bat above the back shoulder, and a downswing that brings the

 doi:10.20944/preprints202604.1970.v1

bat through the line of the ball. The differences among these strokes reside in subtle parametric variations: the precise lateral position of the stride (cover drive toward cover region versus straight drive toward mid-off versus on drive toward mid-on), the angle of the bat face at impact (open, square, or closed), the degree of wrist rotation during follow-through, and the timing and magnitude of weight transfer.

The frame-level ResNet50 features, while powerful for object recognition and scene understanding, may not encode these subtle kinematic variations with sufficient fidelity. The features capture high-level semantic information—"there is a batsman playing a shot"—but may not preserve the fine- grained spatial details (bat face angle, wrist position, precise foot placement) that differentiate front-foot strokes. Further- more, the 32-frame temporal sampling, while adequate for capturing the overall stroke arc, may not provide the temporal resolution necessary to resolve rapid kinematic events such as the exact moment of bat-ball contact or the acceleration profile of the downswing.

The Role of Motion Ambiguity: Some front-foot stroke confusions may reflect genuine ambiguity in the visual signal. A cover drive played with controlled power and soft hands may, from a purely visual perspective, be nearly indistinguishable from a firm defensive push. The distinction between these strokes resides partly in the batsman's intent and the outcome (whether runs are scored), information that is not available from the video clip alone. This inherent ambiguity sets a ceiling on achievable classification accuracy and underscores the need for additional information sources—ball trajectory, field placement, match context—to fully disambiguate stroke identity.

E. Limitations of Frame-Level CNN Features for Kinematic Analysis

The error analysis points to a fundamental limitation of our feature extraction pipeline: frame-level CNN features, even from a powerful architecture like ResNet50, are inherently appearance-based and lack explicit motion encoding. The features capture what objects and poses appear in each frame but provide limited information about how those appearances change across frames.

Missing Kinematic Information: Critical kinematic parameters for stroke discrimination—bat velocity profiles, acceleration patterns, bat face angle trajectories, joint angle evolution, weight transfer timing—are not explicitly represented in the 2048-dimensional feature vectors. The LSTM and BiLSTM must infer these motion characteristics indirectly from the sequence of static appearance features, a challenging inductive leap that likely contributes to the observed error patterns.

Comparison with Optical Flow and 3D Convolutions: Alternative feature representations that explicitly encode motion information might better serve this task. Optical flow fields directly measure pixel-level motion between consecutive frames, providing a dense representation of movement that could help distinguish strokes with different velocity profiles. 3D convolutional features, learned end-to-end on video data, can capture spatiotemporal patterns that span multiple frames. However, both approaches come with trade-offs: optical flow computation adds substantial preprocessing overhead, while 3D convolutions require large video datasets for effective training and are computationally intensive.

The Case for Explicit Pose and Trajectory Features: Given the limitations of appearance-based features, a promising direction for improving fine-grained sports action recognition is the integration of explicit kinematic representations. Pose estimation frameworks can extract joint keypoint coordinates, providing a low-dimensional but semantically rich representation of body configuration. Object tracking can recover bat trajectory, yielding direct measurements of the bat path that is central to stroke discrimination. These explicit kinematic features could be fused with appearance-based CNN features to provide complementary information streams, potentially yielding substantial improvements in classification accuracy, particularly for the challenging front-foot stroke cluster.

 doi:10.20944/preprints202604.1970.v1

VI. Limitations and Future Work

While this study provides a rigorous comparative bench- mark and yields valuable insights into temporal modeling for cricket stroke classification, several limitations constrain the scope and generalizability of our findings. Acknowledging these limitations candidly not only strengthens the credibility of our contributions but also illuminates promising directions for future investigation.

A. Dataset Scale and Diversity

The *CricShot10k*dataset, with approximately 10 000 videos distributed across 15 classes, represents a valuable resource for cricket action recognition research. However, the dataset scale is modest by contemporary deep learning standards, where models are routinely trained on millions of examples. This limitation has several consequences:

- **Generalization:** The model's ability to generalize to unseen batsmen, camera angles, match conditions, and stroke variations is constrained by the diversity present in the training set. Performance on out-of-distribution examples may degrade substantially.
- **Transformer Performance:** As discussed extensively, the Transformer's underperformance is directly attributable to the dataset's limited scale relative to the model's capacity and lack of inductive bias. Larger datasets might enable Transformers to realize their potential for this task.
- **Overfitting Risk:** Even the LSTM models, with their favorable inductive biases, operate near the boundary of overfitting given the dataset scale. More aggressive regularization or data augmentation might be necessary for larger models.

Future Direction—Dataset Expansion: Collecting and annotating additional cricket stroke videos would directly address these limitations. Semi-automated annotation pipelines leveraging broadcast metadata, commentary alignment, or weak supervision could facilitate the creation of larger-scale datasets without prohibitive manual annotation costs.

B. Feature Representation Granularity

Our reliance on frame-level ResNet50 features, while enabling controlled comparison and computational efficiency, imposes inherent limitations on the granularity of motion information available to the temporal models.

- **Temporal Resolution:** The 32-frame uniform sampling discards information between sampled frames. Critical kinematic events—the precise moment of bat-ball contact, rapid acceleration phases—may span only a few frames and be inadequately represented.
- **Spatial Granularity:** The 2048-dimensional feature vectors compress spatial information in ways that may dis- card fine-grained cues (bat face angle, wrist position, precise foot placement) essential for discriminating similar strokes.
- **Lack of Explicit Motion:** CNN features are appearance- based; motion must be inferred from appearance changes. This indirect inference is less efficient than explicit motion representations.

Future Direction—Multi-Modal Feature Fusion: Inte- grating complementary feature streams could substantially enhance representational capacity. Candidate modalities include:

- **Pose Keypoints:** Extracting joint coordinates from each frame using pose estimation frameworks (OpenPose, HR-Net, MediaPipe) would provide explicit, low-dimensional kinematic information.
- **Bat Trajectory:** Tracking the bat through the video sequence would yield direct measurements of bat path, velocity, and acceleration—features highly diagnostic of stroke type.
- **Optical Flow:** Dense motion fields between consecutive frames capture pixel-level movement patterns that complement appearance-based features.
- **Ball Trajectory:** The line, length, and trajectory of the delivery contextualize the batsman's stroke choice and execution.

 doi:10.20944/preprints202604.1970.v1

C. Contextual Information Absence

The current classification pipeline operates on isolated video clips of individual strokes, without access to broader contextual information that human experts use for stroke identification.

- **Ball Trajectory Context:** The line (direction) and length (pitch location) of the delivery strongly constrain the set of appropriate strokes. A cover drive is typically played to deliveries outside off stump; a straight drive to deliveries on the stumps. Access to ball trajectory would provide strong priors for stroke classification.
- **Field Placement Context:** The positions of fielders influence stroke selection and execution. A lofted stroke played over the infield is only attempted when the field is up; the same stroke played with fielders on the boundary would be classified differently based on intent.
- **Match Situation Context:** The match format (Test, ODI, T20), innings stage, and score pressure influence stroke aggression and execution. A defensive stroke in a Test match may differ kinematically from a defensive stroke in a T20 powerplay.

Future Direction—Context-Aware Classification: Incorporating contextual features—either as additional input modalities or as post-hoc priors—could improve classification ac- curacy and robustness. Multi-modal architectures that jointly process video, ball tracking data, and match metadata represent a promising direction.

D. Inherent Ambiguity of Visually Similar Strokes

Some classification errors may reflect genuine ambiguity in the visual signal rather than model deficiencies. The distinction between a firm defensive push and a controlled cover drive resides partly in batsman intent and stroke out- come—information not available from the video clip alone.

- **Intent Ambiguity:** Two strokes with identical kinematics may be classified differently based on the batsman's intent: was the batsman trying to score runs (cover drive) or merely defend (defensive stroke)? Intent is not directly observable from kinematics.
- **Outcome Ambiguity:** The same kinematic execution may yield different outcomes depending on the delivery. A stroke that would be a cover drive to a full delivery might be a defensive block to a shorter ball, even with identical bat movement.
- **Technique Variation:** Individual batsmen exhibit idiosyncratic techniques. A cover drive played by one bats- man may kinematically resemble a straight drive played by another. The model must abstract over individual style to extract the invariant stroke category.

Future Direction—Uncertainty Quantification and Reject Options: Rather than forcing a hard classification on ambiguous examples, models could output calibrated probability distributions and abstain from prediction when uncertainty exceeds a threshold. This approach would be more appropriate for deployment in applications where incorrect classifications have meaningful consequences.

E. Architectural Exploration Scope

Our study compared four distinct architectural paradigms (non-temporal, LSTM, BiLSTM, Transformer) but did not exhaust the space of possible temporal modeling approaches.

- **3D Convolutional Networks:** C3D, I3D, and Slow- Fast networks represent an alternative temporal modeling paradigm not evaluated in this study.
- **Attention-Augmented RNNs:** Hybrid architectures that combine recurrence with attention mechanisms might offer a favorable trade-off between inductive bias and representational flexibility.
- **Temporal Convolutional Networks (TCNs):** TCNs use dilated convolutions to capture long-range temporal dependencies without recurrence, offering an alternative to both RNNs and Transformers.
- **Graph Neural Networks:** Pose-based approaches could model the batsman as a graph of joints

 doi:10.20944/preprints202604.1970.v1

and apply graph convolutions to capture kinematic relationships.

Future Direction—Broader Architectural Benchmarking: Extending the comparative framework to include additional temporal modeling paradigms would provide a more complete picture of the architectural landscape for fine-grained video classification.

VII. Conclusion

This paper has presented a comprehensive empirical investigation into the efficacy of temporal deep learning architectures for fine-grained cricket stroke classification using the *CricShot10k*dataset. Through rigorously controlled experimentation comparing non-temporal, recurrent, bidirectional recur- rent, and attention-based models, we have established a clear performance hierarchy, validated key hypotheses regarding the value of temporal modeling, and identified critical limitations of current approaches.

A. Summary of Contributions

Our study makes the following substantive contributions to the field of fine-grained sports action recognition:

1) **Quantification of Temporal Modeling Value:** We have demonstrated that temporal sequence modeling via LSTM yields an 8-percentage-point accuracy improvement over frame-based aggregation, confirming that motion dynamics are essential for discriminating visually similar cricket strokes. This finding provides empirical validation for the widespread adoption of sequence-aware architectures in sports video analysis.
2) **Bidirectional Context Benefit Characterization:** We have shown that bidirectional processing (BiLSTM) pro- vides a modest but statistically significant improvement (2 % absolute, $p = 0.024$) over unidirectional LSTM. McNemar's test confirms that this gain reflects systematic correction of specific misclassifications rather than random variation.
3) **Transformer Limitation Documentation:** We have documented and analyzed the pronounced underperformance of the Transformer architecture in this moderate- data, fine-grained video classification setting. The Transformer's 23.0 % accuracy—substantially below both re- current models and the non-temporal baseline—serves as a cautionary case study regarding the data efficiency limitations of attention-based models when inductive biases are weak.
4) **Error Structure Characterization:** Through per-class F1-score analysis and confusion matrix examination, we have identified a dense confusion cluster among front- foot strokes (cover drive, defensive, down the wicket) where classification accuracy degrades substantially. In contrast, back-foot strokes and strokes with distinctive lateral motion (pull, cut, hook) are classified with significantly higher accuracy.
5) **Feature Limitation Analysis:** We have articulated the inherent limitations of frame-level CNN features for capturing subtle kinematic cues, connecting these limitations to observed error patterns and identifying explicit pose and trajectory features as a promising direction for future improvement.
6) **Reproducible Benchmark Establishment:** We have established a rigorous and reproducible experimental protocol for cricket stroke classification on *CricShot10k*, providing a standardized reference point for future research.

B. Broader Implications

Beyond the specific domain of cricket analytics, our findings carry broader implications for the application of deep learning to fine-grained temporal action recognition:

Architectural Choice Must Align with Data Scale: The Transformer's underperformance underscores that architectural sophistication does not guarantee empirical superiority. In moderate-data regimes, models with strong inductive biases aligned with the structure of the problem domain

 doi:10.20944/preprints202604.1970.v1

often outperform more flexible but data-hungry alternatives. Practitioners should carefully consider dataset scale when selecting temporal modeling architectures.

Temporal Modeling is Essential for Fine-Grained Actions: The substantial performance gap between temporal and non-temporal models reinforces the importance of sequence-aware processing for tasks requiring discrimination among visually similar action categories. Frame-based approaches, despite their simplicity, are fundamentally inadequate for fine grained action recognition.

Appearance Features Alone Are Insufficient: The confusion patterns among front-foot strokes highlight the limitations of pure appearance-based features for distinguishing actions with similar spatial configurations but distinct motion profiles. Explicit motion representations—whether through optical flow, pose trajectories, or 3D convolutions—are likely necessary to achieve high accuracy on fine-grained tasks.

Statistical Validation Strengthens Comparative Studies: Our use of McNemar's test demonstrates the value of statistical significance testing for model comparison. The 2 % BiLSTM improvement would be easy to dismiss as noise without formal validation; the statistical test confirms it as a genuine, reliable effect.

C. Final Remarks

Cricket stroke classification from video stands as a compelling testbed for fine-grained temporal action recognition, challenging models to distinguish among categories that share extensive visual and temporal overlap. Our systematic comparison of temporal modeling architectures establishes that while recurrent networks offer a favorable trade-off between representational capacity and data efficiency, substantial headroom remains for improvement. The confusion patterns we have documented point clearly toward the integration of explicit kinematic features—pose keypoints, bat trajectories, optical flow—as the most promising avenue for advancing the state of the art.

As automated sports analytics systems continue to proliferate, the ability to accurately classify and analyze fine-grained athletic techniques will become increasingly valuable. The benchmark, insights, and future directions articulated in this paper provide a foundation for continued progress toward this goal, with implications extending from cricket to the broader landscape of fine-grained action understanding in sports and beyond.

References

1. A. Author and B. Author, "CricShot10k: A Dataset for Fine-Grained Cricket Stroke Classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2022.
2. H. Wang, A. Kla¨ser, C. Schmid, and C.-L. Liu, "Dense Trajectories and Motion Boundary Descriptors for Action Recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, 2013.
3. H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 3551–3558.
4. K. Simonyan and A. Zisserman, "Two-Stream Convolutional Networks for Action Recognition in Videos," in *Advances in Neural Information Processing Systems*, 2014, pp. 568–576.
5. L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal Segment Networks: Towards Good Practices for Deep Action Recognition," in *European Conference on Computer Vision*, 2016, pp. 20–36.
6. C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional Two- Stream Network Fusion for Video Action Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1933–1941.
7. D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning Spatiotemporal Features with 3D Convolutional Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.
8. J. Carreira and A. Zisserman, "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

 doi:10.20944/preprints202604.1970.v1

9. Z. Qiu, T. Yao, and T. Mei, "Learning Spatio-Temporal Representation with Pseudo-3D Residual Networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.
10. D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A Closer Look at Spatiotemporal Convolutions for Action Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.
11. K. Hara, H. Kataoka, and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6546–6555.
12. J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2625–2634.
13. S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
14. J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond Short Snippets: Deep Networks for Video Classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4694–4702.
15. A. Graves and J. Schmidhuber, "Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures," *Neural Networks*, vol. 18, no. 5-6, pp. 602–610, 2005.
16. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention Is All You Need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.
17. G. Bertasius, H. Wang, and L. Torresani, "Is Space-Time Attention All You Need for Video Understanding?" in *Proceedings of the International Conference on Machine Learning*, 2021.
18. A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucˇic´, and C. Schmid, "ViViT: A Video Vision Transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 6836–6846.
19. Q. McNemar, "Note on the Sampling Error of the Difference Between Correlated Proportions or Percentages," *Psychometrika*, vol. 12, no. 2, pp. 153–157, 1947.
20. K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.