Next-Gen Cricket Scorecard Automation Using Deep Neural Networks

2024-2025

A dissertation submitted to

MEDICAPS UNIVERSITY, INDORE

Towards the partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science & Engineering

Specialization in

Cloud Computing

Submitted By:

Vivek Jindal

Under the guidance of

Dr. Latika Jindal

Department of Computer Science & Engineering

Faculty of Engineering

Medicaps University

Umariya A.B. Road Indore - 453331

Next-Gen Cricket Scorecard Automation Using Deep Neural Networks

2024-2025

A dissertation submitted to

MEDICAPS UNIVERSITY, INDORE

Towards the partial fulfillment of the requirements for the degree of

Master of Technology

in

Computer Science & Engineering

Specialization in

Cloud Computing

Submitted By:

Vivek Jindal

Under the guidance of

Dr. Latika Jindal

Department of Computer Science & Engineering

Faculty of Engineering

Medicaps University

Umariya A.B. Road Indore - 453331

Dedication

I dedicate this thesis to my family, whose unwavering support, love, and encouragement have been the foundation of my academic journey. To my parents, for their endless belief in me and their sacrifices to ensure I had the best opportunities, I owe everything. I also dedicate this work to my mentors and professors, whose guidance and expertise shaped my understanding and approach to the project. Their insightful feedback and encouragement motivated me to push beyond my limits.

Lastly, to the world of cricket, which inspired the topic of this thesis. This work is a tribute to the sport's ever-evolving nature, and my hope is that the advancements in automation and AI will contribute to enhancing the fan experience and analytics in cricket. This work is for those who believe in the power of technology to transform the world we live in.

Vivek Jindal
EN22CS501015
Medicaps University,
Indore, Madhya Pradesh, India

Dissertation Approval

The dissertation report titled as **"Next-Gen Cricket Scorecard Automation Using Deep Neural Networks"** is hereby approved as a creditable study of an engineering application subject carried out and presented in a manner satisfactory to warrant its acceptance as pre-requisite for the degree for which it has been submitted.

It is to be understood that by this approval the undersigned do not endorse or approve any statement made, opinion expressed, or conclusion drawn therein; but approve the "dissertation" only for the purpose for which it has been submitted.

Internal Examiner:	External Examiner:
Name:	Name:
Date:	Date:
Designation:	Designation:
Affiliation:	Affiliation:

Declaration

I hereby declare that the dissertation report titled **"Next-Gen Cricket Scorecard Automation Using Deep Neural Networks"** has been completed and the purpose is to complete the degree of "Master of Technology" in Computer Science & Engineering under the guidance of Prof. (Dr.) Latika Jindal, Associate Professor, Department of Computer Science & Engineering, Faculty of Engineering, Medicaps University, Indore, is an authentic work.

In addition, I declare that the content of this dissertation report has neither been taken from any other source nor have been submitted to any other institute or University for the award of any other degree or diploma.

Vivek Jindal
EN22CS501015
Medicaps University,
Indore, Madhya Pradesh, India

2024-2025

CERTIFICATE

I, **Dr. Latika Jindal**, Associate Professor, certify that the dissertation entitled **"Next-Gen Cricket Scorecard Automation Using Deep Neural Networks"** submitted in fulfillment for the award of the degree of Master of Technology by **Vivek Jindal** is the record carried out by him under my guidance and that the work has not formed the basis of award of any other degree elsewhere.

Dr. Latika Jindal

Associate Professor

Department of Computer Science & Engineering

Medicaps University, Indore

Prof. (Dr.) Ratnesh Litoriya

Professor & HOD

Department of Computer Science & Engineering

Medicaps University, Indore

Prof. (Dr.) Pramod S. Nair

Dean, Faculty of Engineering

Medicaps University, Indore (M.P.)

Acknowledgment

I would like to express my deepest gratitude to Honorable Chancellor, **Shri R.C. Mittal,** who has provided me with every facility to successfully carry out this dissertation, and my profound indebtedness to **Prof. (Dr.) D.K. Patnaik,** Vice Chancellor, and **Prof. (Dr.) Birjashis Patnaik,** Pro-Vice Chancellor, Medicaps University, whose support and enthusiasm has always boosted up my morale. I would also like to thank **Prof. (Dr.) Pramod S. Nair,** Dean, Faculty of Engineering, Medicaps University, for giving me a chance to work on this project. I also thank my Head of the Department **Prof. (Dr.) Ratnesh Litoriya** for his continuous encouragement for the betterment of the dissertation. I would also like extend my heartfelt thanks to **Prof. (Dr.) Kailash Chandra Bandhu**, Academic Head, Department of Computer Science & Engineering, for his constant encouragement, insightful suggestions, and significant support in both my research paper and thesis work. His guidance played a vital role in refining the technical depth and academic rigor of my research. I am also grateful to **Dr. Manoj Kumar Rawat**, Master of Technology Coordinator, Department of Computer Science & Engineering, for his support, coordination, and efforts in ensuring the smooth progression of my academic journey. His timely assistance and encouragement have been immensely valuable.

I would also like to thank my guide, **Dr. Latika Jindal**, Associate Professor, Department of Computer Science & Engineering, for her dedicated support and guidance. She continuously encouraged me and was always willing to assist in any way she could throughout the dissertation.

Vivek Jindal
EN22CS501015
M.Tech 4th Semester
Department of Computer Science & Engineering
Medi-Caps University, Indore

Abstract

This thesis talks about the cricket scorecards automation via machine learning and computer vision techniques, specifically Convolutional Neural Networks (CNNs). Cricket with such a complex scoring and umpire gestures tends to be scored either manually or semi-automatically but is prone to human errors. This presented research work has addressed these underlying issues by developing a completely automated system which takes the real time images of cricket umpire gestures, movements, and produces the precise scorecards. The given approach involves the compilation of a dataset with numerous umpire gestures, preprocessing, and augmentation, which aids in improving model generalization. Various gestures are trained on this dataset and evaluated in a CNN model, which produced a very high classification accuracy of around 97.22% across all types of gestures. The major steps include extracting features with convolutional layers, transfer learning for continued model performance, and hyper-parameter optimization through fine tuning.

The Proposed model demonstrates significant improvement over existing works in precision and recall compared to classical methods due to minimal errors in real-time applications. The model's robustness is validated by manual tests, confirming its usability in live cricket matches. It also discusses the future direction which includes integrating attention-motivation mechanisms for enhanced temporal analysis and the need for an extension of the dataset for a wide variety of gestures and match conditions. The present study not only proposes the state-of-the-art in sports analytics but also exposes the great potential of deep learning in automating intricate real world tasks. Further advances, including real-time deployment and integration with live broadcasting systems, have the potential to revolutionize how cricket matches are interpreted and experienced by audiences across the globe.

Keywords: Cricket Scorecard, Machine learning, Image Processing, Gesture Recognition, Umpire

Contents

List of Figures

List of Tables

List of Abbreviations

2D 2-Dimensional.

3D 3-Dimensional.

AI Artificial Intelligence.

AMR Automatic Modulation Recognition.

API Application Programming Interface.

BSC Balanced Score Card.

CNN Convolutional Neural Network.

CV Computer Vision.

DEA Data Envelopment Analysis.

DNN Deep Neural Network.

FM Frequency Modulation.

FPGA Field Programmable Gate Array.

GPU Graphics processing unit.

GRU Gated Recurrent Unit.

I3D-CNN Inflated 3-Dimensional Convolutional Neural Network.

LBW Leg Before Wicket.

LSTM Long Short-Term Memory.

ML Machine Learning.

NLP Natural Language Processing.

PID Proportional Integral Derivative (Algorithm).

PWM Pulse Width Modulation.

RCNN Region-based Convolutional Neural Network.

ReLU Rectified Linear Unit.

ResNet Residual Network.

RNN Recurrent Neural Network.

SGD Stochastic Gradient Descent.

SLR Sign Language Recognition.

SNR Signal-to-Noise Ratio.

SVM Support Vector Machine.

Chapter 1

Introduction

CHAPTER 1

Introduction

1.1 Overview of Automated Systems and Machine Learning in Sports

Automated systems and machine learning [ML] methodologies are progressively changing and re-shaping the field of sports by improving the efficiency, preventing harm, and some possible injuries, and by generating innovative betting stratergies. These digital solutions utilizes large amounts of data to deliver real-time analysis and feedback, improve the training, and also enhance the decision-making process for that particular sport.

Wearable device sensors such as; accelerometers, gyrosocpes, etc., collects the data that the machine learning [ML] model analyzes to provide the real time feedbacks on the performance of players [5]. Artificial intelligence (AI) technologies also enables the development of virtal coaching platforms which helps in enabling of the customized, and personalized training sessions for the players as per their needs and requirements [6].

The present-day status of sports analytics sees AI in a new light for improving the quality and speed of gameplay insights and scoring methodologies. Among the various sports, cricket has demonstrated to be difficult owing to its complex scoring system, which heavily relies on understanding player actions and umpire signals. Scoring methods practiced for a long time, most of them manually or semi-automated, are riddled with errors for no better reason than human error and the inherent limitations of rudimentary digital devices. Recent works have attempted to delve into using machine learning (ML) models like logistic regression, Inception V3, and Long Short-Term Memory networks to automate cricket scorecards with mixed results due to poor precision during classification on the intricate visual gestures that define cricket quintessentially. Accordingly, this dissertation endeavours to address the aforementioned issues with a novel approach by proposing an applications-driven deployment of Convolutional Neural Networks (CNNs — a family of DNNs known for their excellence at image recognition tasks).

The motivation behind this research is consequently the compelling reason to go beyond the limited features of available approaches and benefit from the powerful capability of Convolutional neural networks to hierarchically learn, and extract the features from images. This study painstakingly constructs a comprehensive dataset of features of various umpire gestures related to some components of gameplay and develops a CNN model for high efficiency with reduced resource consumption. Various experiments and evaluations have been performed with the proposed model, yielding significant improvements in achieving a level of automation for the cricket scorecard.

Introductory remarks, with cricket being one of the most loved sports globally, involve complex decision-making and collecting substantial data. Machine learning has gained immense attention in automating certain aspects of cricket in recent years, significantly improving accuracy, efficiency, and experiential benefits. Such ML-enabled automated systems in cricket are meant for player performance analytics, umpire decision-making, real-time match analytics, and generating cricket scorecards.

Key advancements of automated systems include ball-tracking systems, such as Hawk-Eye, built on a machine learning basis to compute the trajectory of the ball for the LBW decision, and player tracking systems to analyse player movement and fielding positions. In other words, bat-ball impact detection systems driven by machine learning guide the umpires to take the correct call by analysing audio and video data. These systems incorporate deep learning methods, particularly convolutional neural networks (CNNs), for real-time video and image recognition.

Another big innovation in cricket: automated cricket scorecards, in which ML algorithms detect and classify umpire gestures, player gestures, and scoreboard elements from live feeds in up-to-the-second updates, removing human error and ensuring instantaneous and trustworthy scoring information. Developments in machine learning, mainly computer vision, and deep learning are creating a newer interface as they pave the way for automated systems to efficiently eradicate bias, heal every game of cricket, and introduce microscopically detailed analytical perspectives.

This thesis embodies an original aim toward the automation of cricket scoreboards with the help of Deep Neural Networks. The DNNs are machine-learning algorithms that have received immense success and are performing great in various computer vision tasks. DNNs can learn complex features and patterns from the available data, making this approach a plausible choice for an automatic scoreboard in cricket.

The general idea of this thesis is to create a DNN-based system which can automatically extract & understand cricket scorecards from live video streams. The proposed system will be able to identify and track. Within the past twain decades, machine learning-the powerful tools with respect to analytics-has changed the scenario for the entire sports industry in performance, strategy, and fan engagement. The first steps in ML in sports go back to the early 2000s, with basic statistical models used to analyze performance and optimize team strategies. The game was changed, however, on the shoulders of big data and more advanced algorithms during the 2010s.

1.1.1 Early Adoption (2000–2010)

In the early 2000s, basic predictive models utilized ML for sports like baseball, basketball, and soccer. For example, ”Moneyball,” a book written in 2003 by Michael Lewis, illustrated how the Oakland Athletics used statistical analysis in baseball. This was the era that laid emphasis on using linear regression models to predict player performance and game outcomes, propelling highly data-driven decisions.

1.1.2 Advancements in the 2010s

Everybody could see rapid development in machine learning from 2010 to 2020, when wearable technology networks and sensor data created an ecosystem for its emergence. Football, tennis, and cricket use ML for injury prevention, in-match performance analysis, and tactical analysis. Using support vector machines (SVM), decision trees, & other forms of sophisticated algorithms, the coaches looked at huge amounts of data to fine-tune game strategy. Soccer clubs predict player fatigue and make optimal substitutions using developing ML strategies. NBA teams leverage ML analytics for assessing shooting angles and defensive formations. Computer vision models were further developed for tracking players, optimizing training situations, and aiding referees in decision-making by providing instant replays and video analysis.

1.1.3 Recent Developments (2020–Present)

In the 2020s, a major boom of applications of AI and ML in sports propels deep learning, neural networks, and reinforcement learning. Technologies such as convolutional neural networks (CNNs) and recurrent neural networks (RNNs) are employed in intricate tasks, such as video analysis, player tracking, fan engagement, and so on - all in aid of the remaining applications. For example, the NFL employs ML to analyze game footage about performance, while the NBA utilizes ML for predictive analytics to deliver rich and customized content so as to enhance the fan experience.

Wearing technology and ML combine a real-time insight adaptation of players' biometrics, with the guiding considerations of injury prevention and recovery optimization. Moreover, esports now have a surge of application for ML when it comes to analyzing player behavior and tuning up driven-ai adversaries. Impact on sports business and fan engagement.

Deep learning is an ongoing revolution that is significantly transforming the landscape of sports management through the tool of machine learning. Ticket sales, fan engagement, and personalized marketing campaigns stand testimony to the efficacy of machine learning algorithms. Predictions are made on analyzing fan behavior in order to arrive at pricing for the tickets and to process the marketing campaigns, thus enriching the overall experience for the fans.

1.2 Preliminaries

This created model has been based on principles of Deep Neural Networks, which applied the advanced feature extraction and classification techniques that machine learning controls. This architecture is more than adequate to handle that work-from object detection/classification to medical imaging and agricultural diagnostics.

1.2.1 Deep Neural Networks Foundation

Deep Neural Networks form the backbone of the proposed CNN model, characterized by multiple hidden layers between the input and output layers. Unlike traditional machine learning approaches, DNNs automatically learn hierarchical representations of data through these layers, thereby eliminating the need for manual feature extraction. Each successive layer captures increasingly complex patterns, enabling the network to discern high-level structures within the data.

The emergence of deep learning can attributed to several factors including the large datasets, increased power of computers or built infrastructures, and new advances in algorithmic methods. The use of graphics processing units (GPUs) has played a pivotal role in shortening the time to develop deep neural networks, by providing considerable speedup through parallelization of calculations, along with cutting down the training time from potentially weeks or months, to a few hours. Additionally, the advancements in optimization techniques such as the stochastic gradient descent (SGD) or its variants have made the deep learning algorithms more efficient and scalable, which allows for the training of more complex and sophisticated models than ever before. There have been dramatic advances in deep learning in recent years, with even greater results demonstrating success across multiple applications of deep learning including, but not limited to object detection, image classification, speech recognition, and machine translation. With key advancement breakthroughs, including the success of AlphaGo over human masters in a game of Go, and the development of driverless cars, algorithms have demonstrated the ability of deep learning to tackle difficult real-world problems. However, despite all of these achievements, deep learning still has many challenges to address, including, but not limited to, interpretability, robustness, and generalizability.

Another difficulty facing deep learning is resilience, particularly in adversarial contexts where hostile actors might modify inputs to deceive neural networks. Adversarial attacks have the ability to manipulate deep learning models in a way that they incorrectly classify inputs, even with very little changes that are difficult to notice. This poses a major security threat in applications like autonomous driving and cybersecurity. To address issues regarding robustness, it is necessary to create strategies to defend against adversarial attacks. These

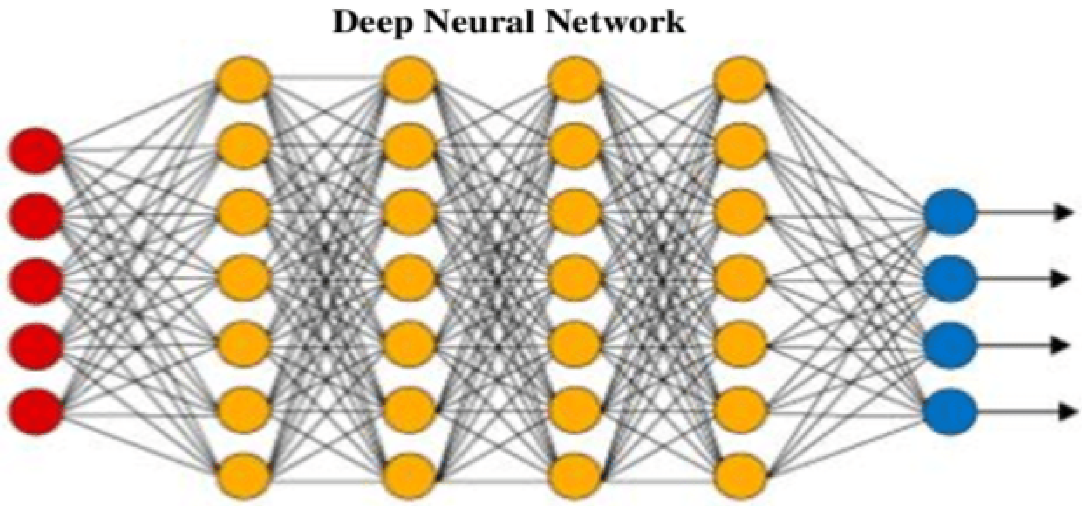


Figure 1.1: Deep Neural Network [1]

strategies include adversarial training, input preprocessing approaches, and robust optimization algorithms.

Moreover, the task of obtaining strong and reliable generalization poses a key obstacle in the field of deep learning. This is because models that are trained on extensive datasets frequently have difficulties in effectively applying their learned knowledge to new and unknown data, or in adapting to alterations in the underlying distribution of the data. Overfitting, a condition when a model becomes too specialized to the training data, can lead to subpar performance and restricted practical use. In order to strengthen the resilience and transferability of deep learning models, researchers are studying regularization approaches, data augmentation tactics, and domain adaption methods to boost generalization.

Not withstanding these obstacles, the outlook for deep learning appears encouraging, as continuous research endeavors are focused on overcoming its constraints and discovering novel capacities. Emerging trends, such as lifelong learning, meta-learning, and neuro symbolic AI, show potential for increasing deep learning and expanding the limits of artificial intelligence. Deep learning, as it progresses, has the capacity to profoundly transform industries, stimulate creativity, and determine the trajectory of technology and society.

Key advantages of employing DNNs include:

- **Hierarchical Learning:** Successive layers capture different levels of abstraction, from basic edges and textures to more complex shapes and patterns.

- **Non-Linear Transformations:** Activation functions such as ReLU (Rectified Linear Unit) introduce non-linearity, enabling the model to learn complex decision boundaries.
- **Deep Feature Representation:** The deep architecture allows for a comprehensive representation of input data, improving model performance on complex tasks.

1.2.2 Machine Learning

The way a Machine Learning (ML) algorithm constructs the model from training data, and subsequently make predictions without explicit programming based systems. In the case of image recognition, A target gesture is by default a discrete sign approximate by the Umpire. Recognizing these discrete signs activates machine learning to its image classification paradigm. There are many machine learning paradigms, with deep learning and feature learning that have made considerable modifications to feature extraction.

The essence of the machine learning paradigm is its ability to draw valuable inferences from data and use those inferences to solve complex problems. In contrast to traditional rule-based programming, which manipulates a computer's behavior based on specific instructions, machine learning algorithms learn from examples and experiences, thus being able to generalize and adapt to new contexts. Typically, the learning process involves training a model to identify statistics in your dataset that include data. The model uses raw data about an activity to spot and map patterns or trends, and can adjust its internal parameters to improve its performance on any specified task. Machine learning algorithms can use complex mathematical techniques, and utilize advanced processing power to detect hidden patterns, generate reliable predictions, and draw actionable insights from large data sets.

A key notion in machine learning is the differentiation between supervised and unsupervised learning. Supervised learning is an algorithm that gains knowledge from labelled data, where each data point is linked to a matching label or output. The objective is to acquire knowledge of a transformation from input characteristics to output classifications, allowing the algorithm to generate forecasts on novel, unobserved data. Supervised learning is commonly used for tasks such as classification, regression, and time-series forecasting. Its purpose is to predict discrete labels or continuous values by analysing input information.

In contrast, unsupervised learning entails acquiring knowledge from unlabelled data, where the algorithm endeavours to discern concealed patterns or structures within the data without explicit instruction. Clustering, dimensionality reduction, and anomaly detection are frequently encountered problems in unsupervised learning. The objective of unsupervised learning is to reveal underlying patterns or group similar data points together based on their intrinsic features. Unsupervised learning methods are essential for exploratory data analysis, data preprocessing, and feature engineering. They allow academics and practitioners to obtain more profound insights into intricate datasets.

Semi-supervised learning is a method that merges aspects of both supervised and unsupervised learning. Semi-supervised learning involves training the algorithm using both labeled and unlabeled data, taking advantage of the large amount of unlabeled data to enhance the limited labeled data. This method is especially beneficial when there is a limited amount of labeled data or when obtaining it is costly. It enables the algorithm to utilize the underlying structure of the data to enhance its performance on a specific task. Semi-supervised learning techniques are utilized in diverse fields such as speech recognition, natural language processing, and computer vision. These algorithms are particularly useful when there is a substantial amount of unlabeled data easily accessible.

Reinforcement learning is a machine learning approach that draws inspiration from behavioral psychology. It involves an agent learning to interact with an environment by taking actions and receiving feedback in the form of rewards or penalties. The objective of the agent is to acquire a policy that optimizes the total reward over a period of time, resulting in the development of intelligent behavior through experimentation and learning from mistakes. Reinforcement learning algorithms are utilized in several fields such as robotics, gaming, and autonomous systems. These algorithms enable agents to acquire the ability to navigate intricate settings, make decisions in the face of ambiguity, and adjust to dynamic situations. Machine learning encounters various obstacles, such as interpretability, robustness, and generalization, despite its potential to bring about significant changes. Deciphering the choices made by intricate machine learning models continues to be a challenging undertaking, giving rise to issues around the openness and responsibility of algorithms. Furthermore, machine learning models are vulnerable to adversarial assaults, in which malicious inputs can manipulate the model and result in unforeseen behaviors. Furthermore, the task of ensuring that

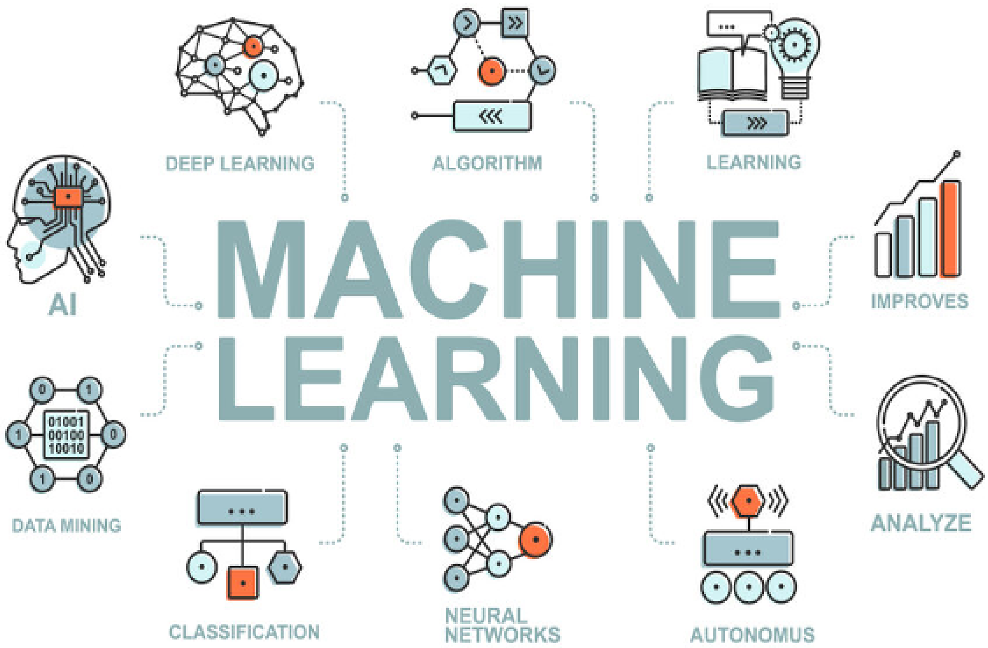


Figure 1.2: Machine Learning [2]

machine learning models have the ability to perform well on new and different data sets and contexts is a significant research challenge. This is because the problems of overfitting and dataset biases can result in subpar performance and incorrect predictions.

In the future, machine learning has great potential, as researchers continue to work on overcoming problems and discovering new capabilities. Emerging technologies like federated learning, meta-learning, and neuro-symbolic AI have promising prospects for enhancing the current level of machine learning and expanding the limits of artificial intelligence. Through the utilization of data and machine learning, we may persist in generating novel ideas, resolving intricate issues, and forging a more promising future for mankind.

1.3 Problem Statement

This thesis presents the solutions to mitigate the limitations of currently available deep learning-based approaches in image classification and improve efficiencies related to image feature classifications via convolutional neural network (CNN). More concretely, the objective of the research is to [7]:

- Find a way to minimize the use of manual scorekeeping methodologies in sports, specifically cricket.
- Find ways to incorporate real-time, automated solutions for data-driven decision making i.e., in this thesis, image analysis via the help of CNN.
- Find ways to improve the accuracy of gesture recognition in dynamic scenarios and real-world environments.

1.4 Role of Deep Learning in Image and Gesture recognition

Image recognition involves identifying and classifying objects, patterns, or features from visual data such as images or video frames. Convolutional Neural Networks (CNNs) are the most commonly used deep learning models for image recognition.

CNNs automatically learn to extract features from raw image data like edges, textures, and shapes.The deeper layers of CNNs can capture higher-level abstractions, such as objects and scenes, making them ideal for complex recognition tasks.

Applications in Cricket:

- Detecting and identifying players, balls, and equipment from live video feeds.
- Recognizing specific events like wickets, boundaries, or player movements.
- Analyzing match footage for player and team performance insights.

1.4.1 Working of CNN

Convolutional Neural Networks (CNNs) play a crucial role in contemporary artificial intelligence, namely in the field of computer vision. These deep learning models excel at processing and analyzing visual data, and can be applied to tasks such as picture recognition, object detection, and segmentation. A Convolutional Neural Network (CNN) is distinguished by its hierarchical arrangement, in which each layer executes distinct operations on the input data. Convolutional layers are the central component of CNNs. These layers use trainable filters to extract features from the input images. These filters detect and extract patterns and structures seen in the data, such as edges, textures, and forms. This allows the network to acquire hierarchical representations of the input. Pooling layers are frequently used after convolutional layers to decrease the size of the feature maps, making them smaller and improving computational efficiency. Pooling methods such as max pooling or average pooling choose the highest or average value from certain areas of the feature maps, respectively. This helps to keep the most important information while getting rid of unnecessary details.

CNNs have a hierarchical structure that allows them to acquire more advanced and intricate characteristics as information flows through the network. Following multiple iterations of convolution and pooling, the feature maps are commonly transformed into a one-dimensional vector and then transmitted across one or more fully connected layers. These layers carry out advanced feature abstraction and associate the acquired features with the output classes or categories. In the CNN, the last layer typically utilizes a softmax activation function to transform the raw output scores into probabilities, which indicate the probability of each class label. During the training phase, Convolutional Neural Networks (CNNs) modify their parameters, specifically the weights and biases, using a technique known as backpropagation. This process entails calculating the gradients of a loss function in relation to the network parameters and modifying the parameters using an optimization method, such as stochastic gradient descent (SGD) or Adam, to reduce the loss and enhance the network's performance.

In addition to conventional computer vision tasks, Convolutional Neural Networks (CNNs) have been utilized in diverse domains such as natural language processing (NLP), speech recognition, and reinforcement learning. CNNs are commonly employed in NLP for tasks such as text categorization, sentiment analysis, and machine translation. CNN-based models has the ability to evaluate and comprehend textual material by capturing semantic linkages and contextual information, hence enabling them to generate precise predictions or classifications. CNNs are used in speech recognition to extract characteristics from audio signals and identify phonemes or words. This enables applications like speech-to-text transcription and voice-controlled assistants. CNNs are employed in reinforcement learning to acquire policies for decision-making tasks, such as game playing and robotic control. CNN-based reinforcement learning agents possess the ability to receive and comprehend visual inputs,

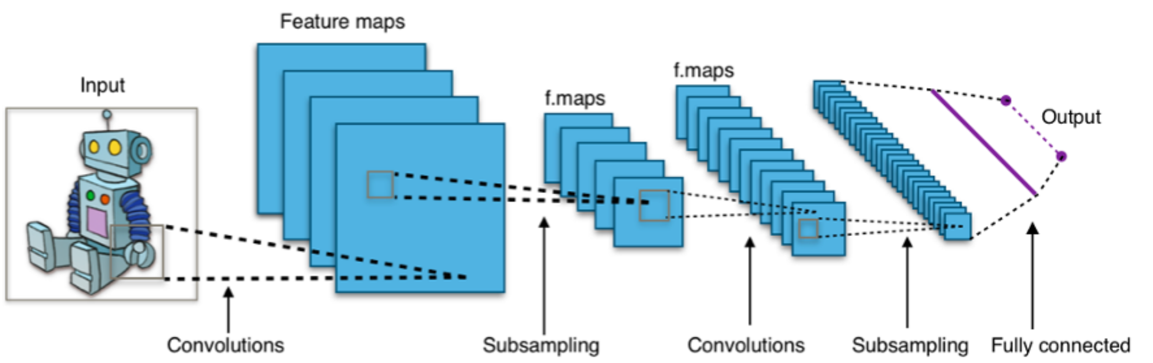


Figure 1.3: Working of Convolutional Neural Network [3]

enabling them to make decisions based on acquired representations in order to accomplish desired objectives.

To summarize, Convolutional Neural Networks (CNNs) have significantly transformed the domain of computer vision and have become essential instruments for evaluating and comprehending visual input. The hierarchical structure of CNNs, together with their ability to learn filters and perform pooling operations, allows them to extract and represent intricate characteristics from unprocessed pixel data. This makes CNNs extremely efficient for tasks like image classification, object identification, and picture segmentation. In addition to computer vision, Convolutional Neural Networks (CNNs) have been successfully applied in other fields including natural language processing, speech recognition, and reinforcement learning, showcasing their adaptability and effectiveness as deep learning models. Although CNNs have achieved considerable success, they still encounter obstacles like as vulnerability to adversarial attacks, the capacity to generalize to unfamiliar data, and the interpretability of the representations they learn. These challenges necessitate additional research and development to be resolved. In general, Convolutional Neural Networks (CNNs) consistently challenge the limits of artificial intelligence and lay the groundwork for novel progress in machine learning and computer vision.

1.4.2 Gesture Recognition

Gesture recognition focuses on interpreting human gestures, such as hand signals, body language, or facial expressions, often from video or image sequences. Deep learning, particularly CNNs, has enhanced the accuracy and speed of gesture recognition systems.

CNNs can analyze video frames to recognize hand movements, body postures, and other signals.These models learn spatial hierarchies in video frames to detect complex gestures.

Applications in Cricket:

- Automatically identifying umpire gestures, such as a raised finger for a wicket or ”no ball” signals.
- Recognizing player celebrations or tactical gestures during a match.
- Updating real-time scorecards and match statistics based on gesture recognition.

1.4.3 Key Benefits of Deep Learning in Image and Gesture Recognition

- **High Accuracy**: Deep learning models offer exceptional precision in recognizing complex visual patterns, even under varying conditions like poor lighting or motion blur.
- **Robustness**: The models handle variations in gestures or images, such as changes in orientation, scale, or occlusion, making them adaptable to real-world scenarios.
- **Minimal Human Intervention:** After initial training on large datasets, deep learning models can operate autonomously, providing real-time updates or analysis with little manual oversight.
- **Real-Time Performance:** The speed of deep learning algorithms allows them to process video feeds or images instantly, delivering immediate insights during fast-paced events like cricket matches.

1.5 Feature Extraction

Feature Extraction is a significant operation in image classification. Here, feature extraction is performed at the level of the convolutional layer. Each of the available Convolutional layers captures more and more complex aspects of image data consecutively. It moves from

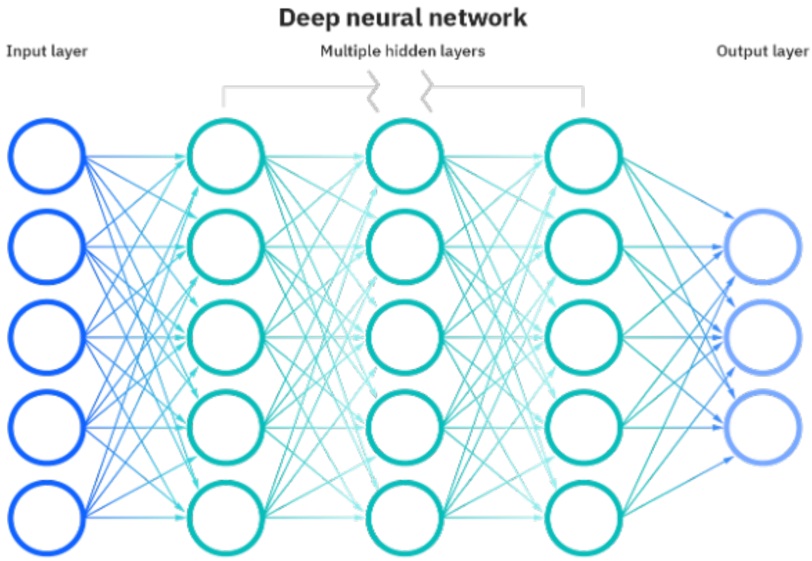


Figure 1.4: Working of Deep Neural Network [4]

basic patterns available in the image to detailed features of gestures done by the Umpire. The task of feature extraction basically defines related data that is available in the given dataset; which makes it a task for a classification model to find the patterns. The task exposes an effective and interesting portion of the provided image in terms of arithmetic feature vectors. Also, applicable when the image size is large, and representation of limited features is required. The main objective of the process of feature extraction from the available images' visual data is to find the essentially required features. The different layers of feature extraction process is can be seen in Figure 1.5.

1.5.1 Transfer Learning

As Figure 1.5 suggests, Transfer learning leverages pre trained models to adapt to new tasks by reusing learned features from large datasets. This reduces training time and improves performance, especially in domains with limited data. Fine-tuning specific layers of the model allows it to specialize in the new task while retaining general patterns from the original dataset.

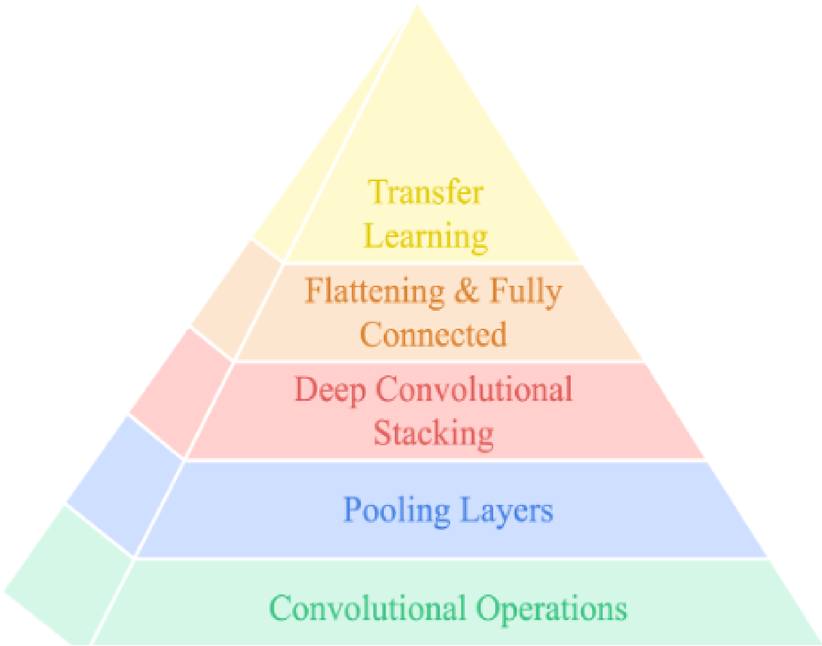


Figure 1.5: Feature Extraction Process.

1.5.2 Flattening Layer

The flattening layer transforms multi-dimensional feature maps from convolutional or pooling layers into a 1D vector. This step bridges the transition to fully connected layers, enabling the neural network to process features linearly for classification or regression tasks.

1.5.3 Fully Connected Layer

Fully connected layers connect all neurons from the previous layer to every neuron in the current layer. They aggregate features extracted by earlier layers and make final predictions using activation functions like ReLU or Softmax, depending on the output requirement (e.g., probabilities or continuous values).

1.5.4 Deep Convolutional Stacking

This involves stacking multiple convolutional layers to progressively learn hierarchical features. Early layers capture simple patterns such as edges, while deeper layers identify complex structures like shapes and objects, enhancing the model's capacity to handle intricate visual tasks.

1.5.5 Pooling Layer

Pooling layers reduce the spatial dimensions of feature maps by summarizing information through operations like max pooling or average pooling. This process decreases computational requirements, prevents overfitting, and provides spatial invariance, making models more efficient and robust to input variations.

Chapter 2

Literature Review

Chapter 2

Literature Review

2.1 Artificial Intelligence

The incorporation of Artificial Intelligence (AI) technology has a significant effect on the conventional approaches to research in a variety of domains [8]. In the field of modern cricket, AI has played a key role in automating several features of the game, especially in the process of decision-making for umpires as well as managing the cricket scoreboards [9]. In an effort to explore further into the latest umpiring methodologies of cricket, some focused research has been conducted; intending to analyze the implementation of artificial intelligence and its resulting impacts on the game of cricket and its participants [10]. This research acknowledges a thorough analysis of available pieces of literature while also collecting qualitative data, hence helping the researchers to thoroughly access the ethical factors related to the adoption of artificial intelligence (AI) into the field of sports [11]. Furthermore, the study also explores into the wider framework of artificial intelligence research; particularly examining the applications and outcomes of tools related to AI like neural networks [12]. Although, the available research studies overviewed provide very little information on the association between sports and artificial intelligence; These studies undoubtedly highlight the increasing impacts of AI in the field of sports. Hence, highlighting the significance of ethical analysis when implementing such tech innovations.

2.2 Machine Learning

Machine Learning [ML] has always been the area of significant studies and research inside the domain of cricket statistics. A number of research articles have explored the applications or implementations of machine learning approaches for the analysis of various cricket matches and anticipating the outcomes of the match. Related to this; a research approach mentioned an innovative approach of hybrid machine learning was presented explicitly for the video summarization task in the sports of cricket. This novel approach used, validated an outstanding accuracy in its ability to identify some major events of the game such as, sixes, fours, and wickets from the available video recordings [13] of cricket matches [14].

Furthermore, another team of researchers worked and developed an algorithm to optimize and automate the process of decision making of umpires of cricket by transcribing the hand gestures shown by them. This suggested algorithm has competently eradicated the need for manual scorecard updations process and has successfully reduced the overall duration of the game to an extent [12]. Another approach for generating automating highlights of cricket gameplay was also introduced. This methodology involves the use of score extraction and techniques of action recognition to help in detection of vital events. This proposed approach was proved to be highly effective in collecting and capturing the most considerable moments of a cricket match [15].

Additionally, a separate group of researchers have conducted a detailed study; in which they evaluated various machine learning (ML) approaches to analyse the pitch on which the cricket match is going to be played, this is done to predict the possible outcomes of the cricket match. The conclusion of this study suggested that both of the Random Forest algorithms, and Naïve Bayes algorithms have displayed encouraging possibilities in the said domain [16]. On top of that, another team of researchers has refocused their attentions towards the performances of players within the game of cricket by utilizing the machine learning algorithms. Their study provides meaningful revelations that can be applicable in selection of players and can also be helpful for the purpose of strategy development [17].

2.3 Image Processing

The automation of cricket scorecards has gained a remarkable interest in the field of image processing methodologies. One research article introduces a framework which employs visual analysis approaches that identifies signals of Umpires, and thereafter modifies the cricket scorecard accordingly. This pioneering method removes the requirement of manually updation of score keeping and makes the process of scorecard generation more efficient [18].
Automating Scorecard and Commentary Based on Umpire Gesture Recognition

Another investigation conducted by a group of researchers [12] presents a novel approach to comprehensively interpret hand gestures and helps in automating the decisions of the Umpire. By applying this technique, the requirement of operator control (human involvement) inn updating the cricket scorecard can be eliminated [19]. Hence this presented methodology aids the enhancements of processing of images into the domain of scorecard generations in cricket, enabling one more accurate and more efficient system.

In addition to this, one another detailed research literature was also taken in consideration based on the Balanced Score Cards (BSC) highlighting the shortcomings in the already present research works and providing some possible areas for future researchers to investigate [20]. This study draws our attention to the need of more thorough analysis and study on this topic, which ultimately contributes to the in-depth understanding and will help in development of the concept of balanced cricket scorecards.

Apart from those researchers, an additional study which contrasts between different approaches including deep learning methodologies, in the disscussions mentioned; the main objective of this computerized image analysis, also known as automated image processing and analysing [12]. This proposed research serves as a valuable reference for the acquainted of the various approaches and techniques used into the field of Image processing, also providing some valuable information in the prospective benefits of image analysis.

At last, researchers have conducted an assessment of the Field Programmable Gate Array (FPGA) as one of the hardware implementations for the image processing algorithms [21]. This investigation mentions about the possible capabilities and performance of Field Programmable Gate Array in the carrying out (implementation) of the image processing algorithms, while providing some important perspectives for the hardware implementation aspects of this mentioned field.

These above-mentioned studies jointly aids into the improvements of image analysis approaches across different areas, for example, healthcare, traffic management, and sports such as, for cricket; scorecards and automating the decision making process of the Umpire judgements. By employing these visual processing methodologies, various researchers are seeking to improve the effectiveness and the precision of updates in the cricket scorecards, which will help in reducing the necessity of human involvement in the process. Moreover, an extensive research survey [22] mentiones about the gaps and inconsistencies into the field of balanced scorecards. Hence stressing on the importance and necessity of additional research in this field. Comparison of different methodologies mentioned in [20] highlights the diverse approaches available for analyzing digital images, offering valuable insights into the potential applications of image processing. Finally, the performance analysis conducted in [21] provides crucial data on the hardware implementation of FPGA for image processing algorithms. Most often, these investigations contribute to the ongoing progress and comprehension of image processing techniques, pushing the limits of technological advancements in this field.

2.4 Tensorflow and Keras Platform

TensorFlow and Keras are currently being utilized within the realm of sports with the intention of providing athletes and coaches with immediate and pertinent biomechanical feedback. This feedback serves to assist in the acquisition and refinement of motor skills [23]. By employing the utilization of deep learning techniques, the necessity for an excessive number of sensors required for the monitoring of human motion can be considerably diminished. This subsequently allows for the execution of swifter and more intricate movements, all while ensuring that the athletes' motor capabilities remain unhindered. In a particular study, Sequential Neural Network models were successfully implemented through the utilization of TensorFlow's Keras API. The primary objective of this implementation was to predict significant joint angles based on the respective vertical displacements and velocities exhibited by the waist, wrist, and hip regions [24]. Overall, it is evident that this methodology which is using TensorFlow and Keras helps in accelerating the more efficient analysis of data related to kinematics and kinetics (force and motion). Hence, it is making a significant contribution in the field of players' performance in high level sports.

2.5 Automation

Automation in sports like cricket is a trendy area of focus in the field of research recently. One research paper recommends the use of methodologies like machine learning and visual computing (computer vision), key events detection, and action recognition methodologies for the purpose of generation of self-generating cricket highlights. Another investigation presents an algorithm for the purpose of automating the umpire decisions by translating hand gestures, ultimately helps in eliminating the need for the updations of manually changing the scorecard which resulting in reducing the overall duration of the gameplay [15].

2.6 Survey

The study focuses on cricket activity recognition, proposing an end-to-end CNN-LSTM model designed to address challenges in identifying complex cricket activities. The model employs a combination of time-distributed 2D CNN layers for spatial feature extraction and LSTM layers for temporal sequence learning. The research introduces a dedicated cricket dataset comprising 722 videos across five classes—bowled, cover drive, defense, pull shot, and reverse sweep—collected from online sources. The dataset is complemented by experiments on UCF101 [25], HMDB51, Kinetics, and YouTube Action datasets. The proposed CNN-LSTM model achieved 92.65% accuracy on the cricket dataset, outperforming methods like SimpleRNN and Bi-directional LSTM. On benchmark datasets, the model's accuracy ranged from 71.10% (YouTube) to 90.03% (UCF101). Evaluation metrics on the cricket dataset indicated strong performance, with class-level precision, recall, and F1-scores of up to 99%. Challenges like lower accuracy in the reverse sweep class (79%) were attributed to its complex nature. The results highlight the model's robustness and generalization potential across varied datasets [26].

The given research study shows the development of a three-wheeled bowling machine for cricket that is integrated with a fuzzy logic controller to replicate the deliveries of different styles of ball throws. The machine has utilized a Mamdani type of fuzzy inference system [27]; which helps in generating the frequency modulation signals, which is also known as pulse-width modulation (PWM) signals for three motors which controls the speed and spins both laterally and vertically. This study evaluates seven layouts with different styles of spins and different and speeds on a cricket ball. The dataset used to test this model's efficiency is included with some experimental data recorded with the help of a high-speed camera. The machine achieved a maximum velocity ratio of 92.9% at a low-speed setting and a spin ratio of 30% for vertical spin, although these ratios decreased at higher speeds. The average error for ball projection along the pitch varied between 0.407 m and 3.861 m, attributed to mechanical inconsistencies at higher angular speeds. The study demonstrates the machine's potential for controlled, repeatable deliveries in cricket training [28].

The paper details the creation of a cricket bowling machine with a tri-wheel configuration and a fuzzy logic controller, achieving a delivery precision within 0.423 meters at 1000 RPM. The machine's accuracy is inversely proportional to the wheel speed, with a notable increase in error at higher speeds. It also allows for the adjustment of vertical and horizontal spin, affecting the ball's trajectory over a range of 1.7 to 3.2 meters along the pitch [28].

The research demonstrates a video-driven methodology for the purpose of recognizing significant moments in the sports of cricket by the help of analyzing of signals shown by the Umpire. This uses the inflated 3-dimensional convolutional Neural Network (I3D-CNN) model. The dataset used for this analysis consists of 504 video clips of six different signals (for example, six, wide, out etc.) and around 2000 different images of umpires to detect the umpires' frames. The process of pre-processing helped the resources in ensuring the uniformity of the input by standardizing the size of frames at 224*224 pixels. The model that is presented is optimized by using the Kinetic-600 pretrained weights, which helped in achieving a testing accuracy of 86.14% for the identification of umpire signals, and 97.76% for the detection of frames of umpires. The event detection model was tested on a 15-minute-long cricket video showed a precision of 95.23%, recall value of 86.95%, and FI scores of 90.90%. This presented methodology has surpassed an image-based identification system, by the help of advantages like capturing the motions and for better precision rules. The results shown in paper highlights the capabilities of video-based convolutional neural networks (CNNs) in the process of automating the event detection for the cricket match analysis [29].

The given research paper discusses about a model prepared on Random Forest to categorise images that are shot into a cricket game in six categories, that are being based on the postuers of the batsman. The team of researchers use the Shotnet dataset. The data set was having 3600 images of cricket shots; out of these 3600 images only 2209 images were processed. This is due to the issues of images like noise issues, blurred images, or dark visuals. The keypoints of the body were extracted by utilizing the Mediapipe BlazePose model [30]. 15 of the most relevant key points were used to classify the proposed model (Random Forest model) which achieved an F-1 score of 87% which outperforms the earlier CNN based Shot-net model by 5%. Furthermore, a correspondent assessment method was also introduced to compare and contrast between a normal cricket player's shot with those of players who plays international cricket matches. This enhances the functionality in the domain of performance tracking. This provided solution was also implemented in a mobile application, which helps the user to analyse, enhance and track their performance in batting over the time [31].

The team of researchers presented a live (real-time) face detection, and face recognition system of the players for the sports of cricket. For the detection of players, the AdaBoost algorithm was used and for the face recognition, PAL-based recognition model was utilized. The Proposed System was trained and tested over the dataset of 850 images, which covers different-different scenarios like various poses, obstructions, the changes in lightening etc. The results of the presented method shows the accuracy of 93.5% for the module of player detection, and 93.33% accuracy for the face recognition module. Model shows its robustness by detecting the images of smaller sizes of 20*30 pixels [32].

The study focuses on automatic modulation recognition (AMR) of hybrid modulation signals using a Convolutional Neural Network (CNN) architecture. Four types of hybrid modulations, namely FM-BPSK, FM-QPSK, FM-8PSK, and FM-2FSK, are analysed. The dataset consists of 10,000 training samples and 1,000 test samples for each modulation type with varying signal-to-noise ratios (SNR) and frequency modulation (FM) indices. The proposed convolutional neural network model uses the phase component (P) of the signals for the purpose of input. This approach has exceeded performance of the already available traditional input methodologies like, amplitude-phase (A/P) method, and in-phase quadrature (I/Q) inputs. The results of the proposed model have discussed about achieving an accuracy of 99.8% when the signal-to-noise ratio is 10 dB, and the frequency modulation is 1. Hence, it considerably surpasses already available methods. The accuracy of this investigated model stays above 96% for the most of modulation types [33].

The literary investigation of the article presumably entailed an all-encompassing assessment of prevailing methodologies and technologies in the realm of gesture cognition. Albeit the explicit particulars of said literary investigation elude us in the abstract, it is plausible to deduce that the authors must have delved into the conventional convolutional neural network (CNN) architectures, which are conventionally employed for tasks pertaining to image and pattern cognition. They would have also scrutinized the constraints of these networks, such as the intricacy of feature extraction, the time-intensive nature of network training, and setbacks in terms of cognition precision. Furthermore, the writers would have examined the relevancy of support vector machines (SVM) for tasks related to classification and their efficacy compared to the Softmax classifier typically used in CNN based models. The investigation would have encompassed research on methodologies to forestall overfitting, such as the utilization of batch normalization (BN) layers, as well as approaches to curtail network parameters, such as substituting the fully connected layer with an adaptive global average pool (GAP) layer. This preliminary research would have laid the groundwork for the authors to posit their optimization algorithm that amalgamates an enhanced CNN with an SVM classifier to augment gesture cognition performance [34].

The research paper proposes a method for selecting Indian cricket team players by evaluating their efficiency using Data Envelopment Analysis. DEA is applied to assess past performances, generating efficiency scores by comparing a set of comparable units through the ratio of weighted outputs to weighted inputs. The model considers multiple factors and converts fractional into linear programming problems to calculate player efficiency. The study shows that the team formed by DEA closely matches the actual team selected by the cricket board, indicating the model's effectiveness in strategic player selection [35].

The Author explores the application of computer intelligence in analyzing and predicting cricket match outcomes, acknowledging cricket's unpredictability and complexity. It examines existing research, identifying principal challenges in three main areas: player performance analysis, match simulation, and team selection. Different approaches and solutions are evaluated with the aim to document their strengths and weaknesses. The work is positioned as ongoing research intending to address the gaps and drawbacks uncovered, thereby aiding future studies in cricket data analysis and match prediction [36].

Another team of researchers explores a new system for controlling a small ball on a smooth square plate using the computer vision (CV), and a Proportional Integral Derivative (PID) algorithm. The system uses an python-powered machine vision modules camera (OpenMV camera) for the acquisition of image and an STM32 microcontroller for the data processing and control. The Image data that is collected from the camera is processed to identify the coordinates of the ball; which are adjusted relative to the target position. The Proportional Integral Derivative (PID) algorithm ensures the precise steering control of the plate to move the ball as intended. The dataset used is comprised of real-time images that are captured by the OpenMV camera, with a resolution of 360x360 pixels to suit the dimensions of the plate. Extensive testing demonstrated effective control, and achieving deviations of as low as 2–5 pixels under the optimal conditions. The system successfully balanced the ball and guided it to the predefined targets, although challenges like lighting variations were also noted [37].

The paper introduces an augmented reality-based system for cricket that automates ball tracking and decision-making processes for no-ball and wide-ball scenarios. The model leverages video processing techniques such as HSV color space conversion, edge detection, and Hough Line Transform, alongside a Douglas-Peucker algorithm for trajectory and feature analysis. The dataset comprises video feeds capturing cricket scenarios under varying conditions; however, specifics about size or source are not provided. The results showcase the model's capability to accurately detect no-balls and wide-balls based on geometrical and trajectory analyses. Quantitative accuracy metrics were not explicitly stated, but the system's performance demonstrates its utility for umpiring and player training [38].

The research project has developed a hand gesture based Sign Language Recognition (SLR) system using AlexNet Convolutional Neural Network (CNN) and MediaPipe for detecting and tracking hand landmarks. It has its self-made dataset of 15 classes with a total of 2,536 images, filmed using a video camera under stable conditions. The images are of the resolution of 227×227 pixels and some like commonly used words in Indian Sign Language (ISL), such as "yes," "stop," and "hello." Seventy-five percent of the data are used for training, whereas 25 % for testing. The proposed model has a high accuracy of 98.9 percent, and the other performance metrics are precision (99.33%), recall (99.07%), and F1 score (99.16%). The results confirm that the model performs very well translating hand gestures into text which is important in the normal implementation of SLR [39].

The research paper, titled "Automated Third Umpire Decision Making in Cricket Using Machine Learning Techniques," evaluates machine learning models for automating decision-making in cricket, focusing on run-outs and no-balls. Two approaches were compared: Support Vector Machine (SVM) and Convolutional Neural Networks (CNN). The dataset used consisted of over 2000 images of cricket scenarios, with synthetic data augmenting real samples. While SVM managed an accuracy of 53%, the CNN model using the VGG16 architecture performed significantly better at 88.89% accuracy for run-outs and 89.29% for no-balls Among other key metrics, precision, recall, and F1 scores all showed that CNN had a definite advantage, namely with the precision of 0.81 now higher than that of SVM (0.54) and recall of 0.80 also on the higher side as compared to SVM (0.52). The level of success of the CNN model arises from its advanced image processing techniques and neural network architecture, reinforcing its role in minimizing human error and with that, rendering fairness to the game [40].

Another team of researchers involves a event detection approaches and video summarization in cricket. The models described here include mainly three sets of approaches Support Vector Machines (SVM), Convolutional Neural Networks (CNNs), and other machine learning methods used in shot boundary detection and classification-k-means clustering. Various datasets including features such as shot boundaries, replay segments, and audio-visual content in cricket have been summarized. The accuracy levels are mainly discussed in terms of detection efficiency-for example, shot classification using pre-trained AlexNet CNN achieved the detection efficiency of up to 99.26%, while shot boundary detection techniques showed a maximum precision of up to 95%. The result emphasizes the role of domain-specific context in cricket video summarization-instead of dealing with video sequences, generating short semantically informative summaries has seen a dramatic improvement because of it. Limitations include high computational costs and sensitivity to variations in both visual and text data. Future work emphasizes the integration of multimodal features and the use of advanced neural networks for enhanced event detection and broad applicability to sports analysis [41].

This paper presents a hybrid architecture, CricShotClassify, introduced for cricket batting shot classification by employing CNN and GRU in conjunction. Ten different cricket shots with variations in lighting and conditions were recorded in a dataset created by the authors themselves, referred to as CricShot10. Multiple models were developed and evaluated, including custom CDC-GRU and DCNN-GRU architectures, while using transfer learning techniques with pretrained models like VGG16, InceptionV3, Xception, and DenseNet169. VGG16-GRU gave the best results, attaining accuracy amounts of 93% when the weights of VGG16 for layers 4 and 8 were fine-tuned. The comparative results showed the benefits of transfer learning: the VGG16-GRU consistently outperformed, among other candidates, several custom and transfer learning approaches. However, the successful model faced challenges of misclassifications, which were caused due to confusion between visually similar shots, in particular pull and hook shots [42].

Shot-Net is a 13-layer convolutional neural network (CNN) model for categorizing cricket shots into Cut Shot, Cover Drive, Straight Drive, Pull Shot, Scoop Shot, and Leg Glance. The size of the dataset comprised is 3600 grayscale images divided among the classes, where 80% were set aside for training and 20% for testing. The rest of the rotation, noise addition, and shading are also used for image augmentation. The architecture consists of three convolutional layers of 32, 64, 128 filters each, three max-pooling layers having 2×2 size, four dropout layers of 20%, and two dense layers, and the training is being done using Adam optimizer. An accuracy of 80% was recorded, with a mean Precision, Recall, and F1-score of 0.80, 0.79, and 0.79, respectively, showing that some obstacles still need difficulties with the classification of similar-looking shots, hence further improvements are necessary for future application on 3D depth-based classification with improved algorithms [43].

Table 2.1: Literature Review Summary Table

S No.	Methodology	Result	Advantage	Disadvantage	Future Scope
1.	CNN-LSTM model for cricket activity recognition [26].	Achieved 92.65% accuracy on cricket dataset.	Effectively captures spatial and temporal features, outperforms prior methods.	Limited performance in complex activities like reverse sweep; computationally intensive.	Extend dataset to include more activity classes; incorporate attention mechanisms for real-time analysis.

S No.	Methodology	Result	Advantage	Disadvantage	Future Scope
2.	Three-wheel cricket bowling machine with fuzzy logic controller [28].	Maximum velocity ratio: 92.9%; Spin ratio: 30%.	High precision, repeatable training; simulates varied ball types.	Mechanical errors at high angular speeds; decreased accuracy for horizontal spin.	Improve mechanical robustness; add randomness to ball patterns.
3.	Video-based event detection using I3D CNN [29].	97.76% (frame detection), 86.14% (signal recognition); F1: 90.9%.	Captures motion, high precision, better than image-based methods.	Misclassifications due to overlapping gestures or rare signals.	Expand dataset, incorporate new signals, generate text commentary and highlights.
4.	Random Forest with 15 body keypoints using BlazePose [31].	Achieved 87% F1-score, 5% improvement over CNN.	Accurate classification of cricket shot images.	Limited dataset due to noise.	Expansion to cricket videos and inclusion of more diverse cricket shot types in the dataset.

S No.	Methodology	Result	Advantage	Disadvantage	Future Scope
5.	AdaBoost and PAL-based face recognition [32].	93.33% accuracy for face recognition.	Real-time performance, robust under variation.	Struggles with extreme occlusions.	Expansion to 3D models, multi-angle views, and integration with player tracking and analytics.
6.	CNN using phase (P) as input [33].	99.8% accuracy at 10 dB SNR.	Higher accuracy and simpler preprocessing.	Reduced performance with A/P or I/Q inputs.	Extend to more modulation schemes and investigate robustness under higher interference levels.
7.	Improved CNN with SVM [34].	97.8% accuracy.	High accuracy in gesture recognition.	Dataset details not provided.	Extend to real-time applications.
8.	Data Envelopment Analysis (DEA) [35].	Players like Kohli, Ashwin scored efficiency 1; team matches BCCI's selection.	Quantifies efficiency objectively; aligns with expert decisions.	Limited by historical data availability.	Expand to include new players and T20/ODI formats.

S No.	Methodology	Result	Advantage	Disadvantage	Future Scope
9.	Naïve Bayes, Linear Regression, SVM with historical and social media data [36].	Up to 91% accuracy.	High accuracy with feature engineering.	Insufficient data for new players, dynamic game conditions.	Ensemble methods combining approaches.
10.	PID Algorithm, OpenMV Camera, STM32 [37].	Deviations reduced to 2–5 pixels.	Cost-effective, precise control.	Sensitive to lighting variations.	Adapt for broader use cases like robotics.
11.	Automated ball tracking and decision-making using HSV conversion, edge detection, and geometric analysis [38].	Accurate detection of no-balls and wide-balls.	Enhanced umpiring decisions and player training.	Dataset details and quantitative accuracy metrics not provided.	Extending to real-time analysis and other sports scenarios.
12.	AlexNet CNN with MediaPipe [39].	98.9% accuracy, Precision: 99.33%.	High accuracy, robust hand tracking.	Limited to 15 classes.	Extend dataset and ISL vocabulary.

S No.	Methodology	Result	Advantage	Disadvantage	Future Scope
13.	SVM and CNN (VGG16 Architecture) [40].	CNN achieved 88.89% (run-outs), 89.29% (no-balls); SVM achieved 53%.	CNN provided higher accuracy and better performance metrics.	SVM underperformed, CNN is computationally intensive.	Utilize real datasets for improved accuracy and extend to other cricket scenarios.
14.	SVM, CNN, k-means clustering for cricket video analysis [41].	Shot classification achieved 99.26% accuracy.	High precision in event detection.	Computationally intensive, domain-specific requirements.	Integrating multimodal features for broader applications.
15.	CNN-GRU, DCNN-GRU, Transfer Learning (VGG16, etc.) [42].	CNN-GRU: 82%, DCNN-GRU: 83%, VGG16-GRU: 93%.	Effective feature extraction and temporal data handling.	Computationally expensive, dataset-specific limitations.	Expand dataset, optimize architectures, and include diverse cricket shot categories.

S No.	Methodology	Result	Advantage	Disadvantage	Future Scope
16.	CNN with 13 layers, augmented grayscale dataset [43].	80% accuracy, F1-score: 0.79.	Effective for small-scale cricket shot classification.	Misclassification of visually similar shots.	Enhance model with 3D depth-based classification and advanced algorithms to improve accuracy and robustness.

Chapter 3

Proposed Methodology

Chapter 3

Proposed Methodology

The Machine Learning (ML) is one of the technologies that are very frequently used nowadays. In this thesis, the implementation of machine learning (ML) methodologies like convolutional neural networks (CNN) and ReLu are the primary areas of research in the field of sports analytics. The system that is used to implement the proposed machine learning (ML) model for scorecard automation is demonstrated in this chapter.
The Listing 3.1, presents the pseudo code of the proposed CNN Model, while the Figure 3.3, showcases the visual pyramid of the system.

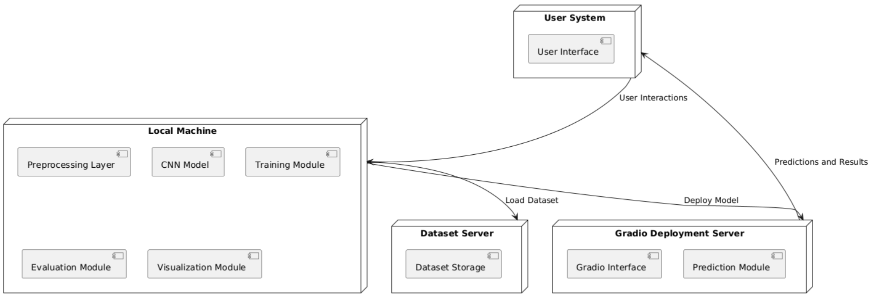


Figure 3.1: Deployment diagram

The Figure 3.1, deployment diagram is representing the architecture of the system for the implementation of the proposed CNN-based model that is described in the pseudo code (Listing 3.1). As per the diagram, the User System is acting as the interface of client where users can interact with the application for the tasks, such as, predictions, and results visualization.

3.1 System Overview

The proposed next-generation automation scorecard system using CNNs is expected to revolutionise the conventional scorekeeping approaches through outstanding computer vision techniques [44] to bring an efficient, accurate, multipurpose, and real-time system for cricket scorecard automation, thus eliminating human errors and improving spectator experience. The system would, relevantly, use CNNs to quickly and accurately recognize player actions, update the scores, and display statistics through a real-time update. This innovation is going to not only provide an easier way to score but also maintain an interesting viewing experience for the fans.

3.2 Proposed System Architecture

The architecture as shown in Figure 3.2 consists of the following four major components, which operate in perfect harmony to ensure seamless data transfer and system functionality. Each component is thus dedicated to a single function and gives rise to a pipeline, wherein raw images are input as data for online log-a-systems. The system serves up-to-date score-cards, with a view to delivering in-depth analysis and insights to the viewers and stakeholders alike.

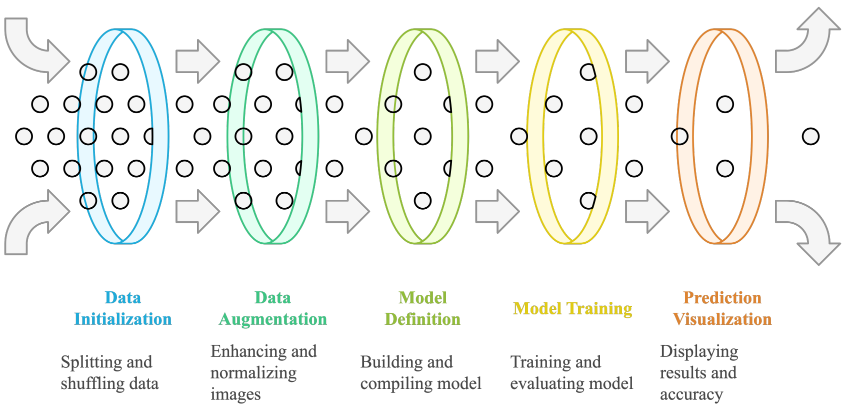


Figure 3.2: Proposed system architecture overview

3.2.1 Pseudo Code

Listing 3.1: Pseudo code for CNN-based model implementation

```
#Import required libraries
Import tensorflow, keras (models, layers)
Import matplotlib.pyplot

# Set image size and batch size
```

```
Set image_size = 256
Set batch_size = 32

# Load dataset from directory
Load dataset from "score" directory with shuffle enabled
and specified image size

# Display class names
Get and display class names from dataset

# Split dataset into training, validation, and test sets
Set train_size = 0.8
Set val_size = 0.1
Take first portion of dataset as train_ds (80%)
Skip the next portion and take as test_ds
Take a portion from test_ds as val_ds (10%)

# Function to get data splits
Define function get_data(ds, train_split, val_split,
test_split, shuffle, shuffle_size)
    Return train_ds, val_ds, test_ds

Call get_data to obtain train_ds, val_ds, test_ds
Cache, shuffle, and prefetch training and validation datasets

# Preprocessing layers
Define resize_and_rescale layer
Define data_augmentation layer for random flipping and rotation

# Build CNN model
Set input_shape and number of classes
```

```
Define model as Sequential with:
    – Resize and rescale layer
    – Data augmentation layer
    – Convolutional and pooling layers
    – Flatten layer
    – Dense layers with softmax activation

Display model summary

# Compile the model
Compile model with:
    – Adamax optimizer
    – Sparse categorical cross–entropy loss
    – Accuracy metric

# Train the model
Train model on train_ds for 50 epochs, with validation on val_ds
Store training history

# Evaluate the model
Evaluate model on test_ds and display scores

# Visualize results
Plot training and validation accuracy over epochs
Plot training and validation loss over epochs

# Define function to predict images
Define function predict_image(img)
    Preprocess and reshape input image
    Predict class using trained model
    Return predicted class name
```

```
# Set up Gradio interface for prediction
Import gradio
Create Gradio interface with:
    - predict_image function
    - Image input
    - Label output
Launch interface
```

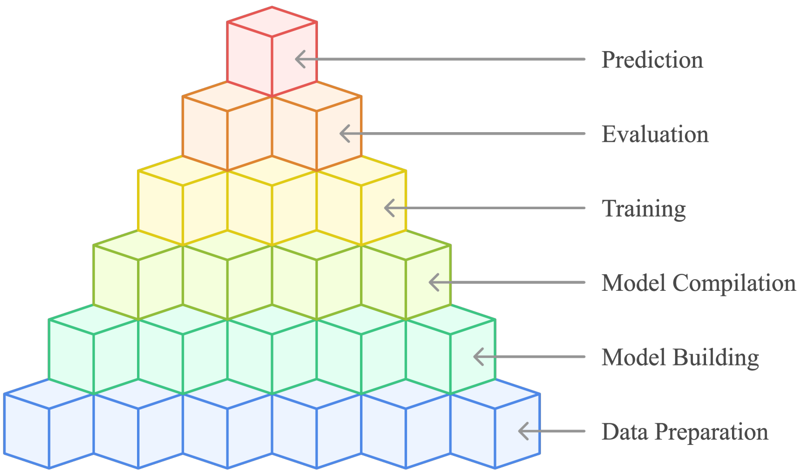


Figure 3.3: Proposed CNN Model Workflow Pyramid

3.2.2 Data Initialization

As shown in Figure 3.4, To start the process of data initialization (DI) for generating the prediction model, The first fragment of data is divided into five sub-parts; the working of these are explained further below.

The Figure 3.5 shows a detailed flow of how data is processed to generate the initial dataset for further process of scorecard generation.

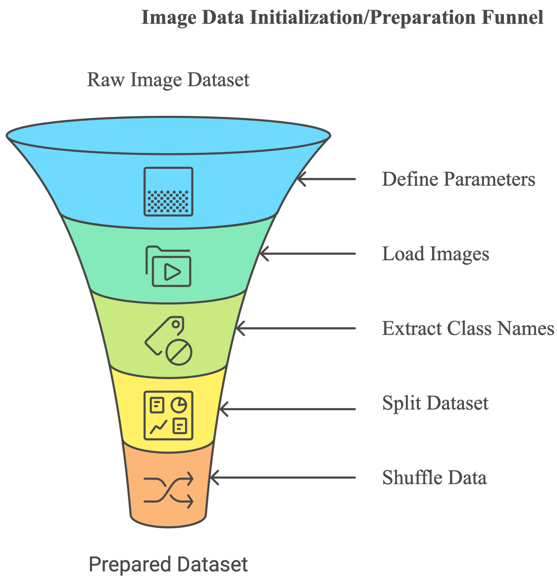


Figure 3.4: Data initialization/preparation funnel

3.2.2.1 Defining Image and Batch Size

Standardizing image size ensures uniformity in input dimensions for the CNN model, which is a crucial part of training and inference. Defining a batch size helps in determining how many samples are processed at once, influencing the system's memory usage and training speed. In cricket scorecard automation, image frames are extracted and converted, and resized to a fixed-dimension image (e.g., 224x224 pixels for the proposed model). This resizing simplifies computation and ensures compatibility with the CNN architecture.

3.2.2.2 Loading Images from the directory

The system retrieves raw images from a directory that houses organized datasets. This structure usually follows a hierarchy where subfolders represent class labels (e.g., actions like batting, bowling, or umpire signals). Images representing cricket-specific events are stored in a structured format. Loading these images efficiently ensures a streamlined pipeline for training and testing the CNN model. Proper organization accelerates the process of associating images with corresponding labels.

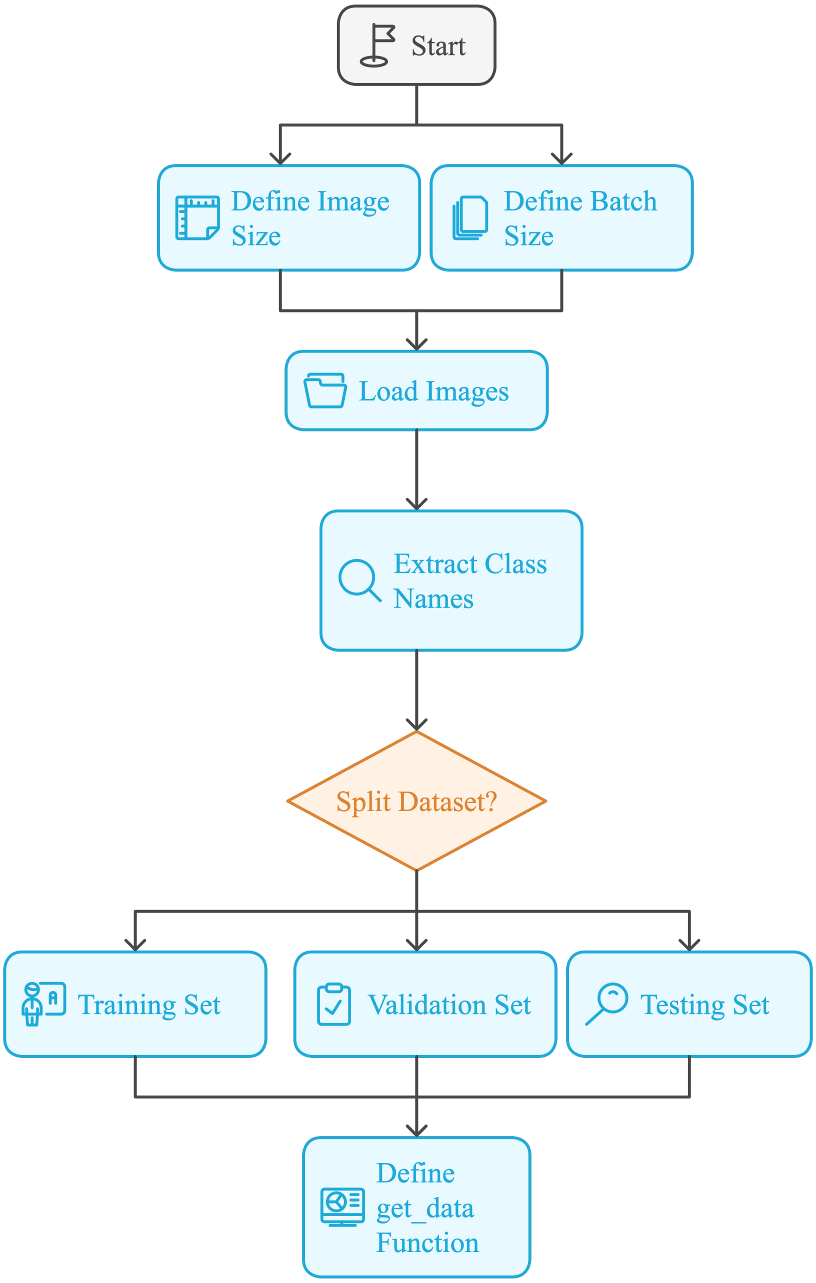


Figure 3.5: data initialization flow

3.2.2.3 Extract Class Names from the Dataset

The next step involves defining the different categories or labels with which the model needs to classify. The classes we defined in this study include boundary, wicket, no-ball, and dot-ball. The purpose of extracting these class names from the dataset is to enable the system to get awareness of all cricket scenarios it will recognize and predict. This is very important for model training and validation, enabling accurate detection and classification of match scenarios.

3.2.2.4 Split the Dataset into Training, Validation, and Testing Sets

Splitting the dataset makes certain that trained, validated, and tested subsets are entirely distinct, preventing any kind of overfitting and assessing the generalizability of the model. Some typical splits include, among others, a ratio of 70-15-15, which means 70% for training, 15% for validation, and the balance 15% for testing. In the automation of the cricket scorecard, the training set drills the CNN model to recognize a pattern of events that occurred in games. The validation set helps tune the hyperparameters of the CNN experience, and the test set shows how reliably the final model performs in actual practice. This partitioning offers added robustness and reliability.

3.2.2.5 Define a Function get_data to Perform the Splitting with Optional Shuffling

This function automatically divides datasets into the training, validation, and testing subsets, with an option to shuffle for unbiased distribution. Shuffling guarantees that the subsets represent the diversity of the entire dataset. Along with the Sform data, the importance of shuffling is with respect to cricket data in terms of temporal dependencies among events, which, without proper randomization, might insist on biased training of the rest of the data.

3.2.3 Data Augmentation and Pre-processing

Data augmentation and data preprocessing are some of the very important methods for enhancing the performance of prospective machine learning (ML) models; it becomes more particular in domains like sports, where the data available is very limited. The process of data augmentation involves the generation of additional data samples by transformation of already available data, thereby increasing the size, and diversity of the dataset, which ultimately improves the generalization of the machine learning (ML) model [45].

On-the-other-hand, the preprocessing, involves the preparation of raw data for the analysis by cleaning, and converting the processed data into a suitable format for processing. Hence;

these technologies are commonly applied amongst various different fields, which include image processing, fault diagnosis, and other visual field predictions. Figure 3.6 displays the flow of data preprocessing for an easier understanding of the mechanism.

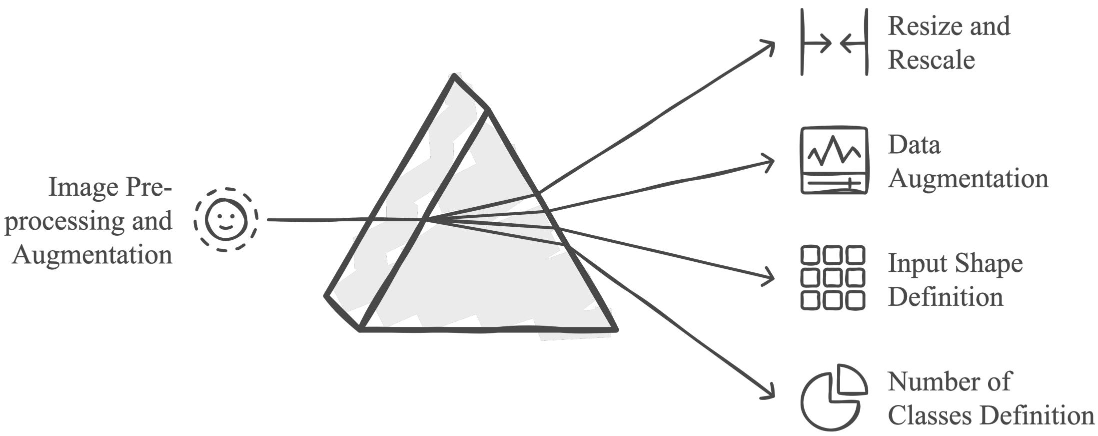


Figure 3.6: Visualizing Image Pre-processing and Augmentation

3.2.3.1 Resizing & Rescaling

As shown in Figure 3.6, The resize and re-scale model is the first process of the pre-processing stage to set the input images for a neural network. Images in datasets are of different sizes and shapes and hence, it can cause inconsistencies while training a model. Simply put, the resizing operation brings all images up to the same size, which is generally taken as the expected dimensions of the model (for example, 224 by 224 pixels for the proposed model). This is a measure taken to ensure the compatibility while reducing the computational load. Besides that, pixel values of normal images usually lie in the range of 0-255. Rescaling these values (hovering between values 0-1) helps in normalizing all of the data and, therefore, improves the numerical stability and speeds up the convergence of optimization algorithms. Some study-engaged pre trained models require mean and standard deviation normalization. This process is important to maintaining integrity across the data set and improving model performance. The Figure 3.7 shows the flow of resizing and re-scaling for the proposed model.

Figure 3.7: Resize & Rescaling flow

3.2.3.2 Data Augmentation Model

The model for data augmentation allows for the training data to be greatly expanded by instilling variations in the input images. Random flipping/rotation shows how it will appear in real-time situations by ensuring that the model can handle susceptibilities in input data. For instance, random flipping flips the image either horizontally or vertically to enable the model to detect objects regardless of their orientation. In the same spirit, random rotation yields varying angles of the object involved, imitating cases when it may appear tilted or oriented at an unusual angle. Besides enriching the dataset, those transformations act as a regularizer so that there is no overfitting, allowing enhancements of the model generalization with respect to unseen data. Data augmentation is of paramount value when working with small data sets, as it effectively enlarges the pool of training data without new labeling.

3.2.3.3 Input Shape

The input shape of a neural network is a primary and crucial parameter that set dimensions for the input data. It usually involves batch size, image dimensions or pixels height and width, and the number of channels. A channel refers to color depth; an RGB image has three channels, while a grayscale image has only one. The batch size affects the number of images fed to the model in one iteration during training or inference, thereby affecting the memory and computation efficiency. The image dimensions provide the resolution at which an image can be though of ideal, retaining detail from the image but also being computationally cheap. A dynamic batch size, given as None, allows leveraging into the deployment process of accepting input for more or less than standard images. Defining the input shape correctly guarantees data compatibility with the model architecture that avoids errors during training and inference.

3.2.3.4 Number of Classes

The number of classes in the dataset defines the distinct types and hence the size of the output layer in a classification model. For example, in the case of CIFAR-10 with ten classes, there would be ten neurons in the output layer corresponding to each class. This is mostly a softmax output layer, calculating the probability distribution over the classes while making sure that the total sum of probabilities would equal one. The number of classes directly influences training, as the model learns to associate features with particular classes. Setting this parameter right is therefore crucial to assure model predictions correlate with both the dataset structure and problem definition.

3.2.4 Model Defination

The sequential model is of great importance in automating the processing of a cricket scorecard (Figure 3.8 displays the overview of the proposed sequential model). It incorporates resize and rescale layers that normalize input image dimensions and normalize pixel values in a consistent preprocessing manner. Data Augmentation layer adds robustness by modeling real-world variations, such as rotations and distortions, while convolutional layers use ReLU activation to extract significant features, such as textual or tabular structures, max pooling layers down-sample these features while preserving important information. The flatten layer changes the output feature maps from the 2D to the 1D format, preparing the data for the fully connected or dense layers such that ReLU captures complex relationships, and softmax acts as the output layer enabling multi-class classification such as identifying player names, scores, or balls faced. Such architecture will give way for generalization of the model for various formats of scorecards for accurate, efficient, and 100% automated scorecard analysis.

3.2.4.1 Create Sequential Model

The Sequential model defines a straightforward approach to create a linear stack of layers where the output of one layer becomes input for the next. This makes it suitable for image classification, where data generally flows from input to output. The intuitive structure also makes implementation and debugging easy, so it is a top contender in research and experimentation. Following a sequential order means the model is computationally efficient,

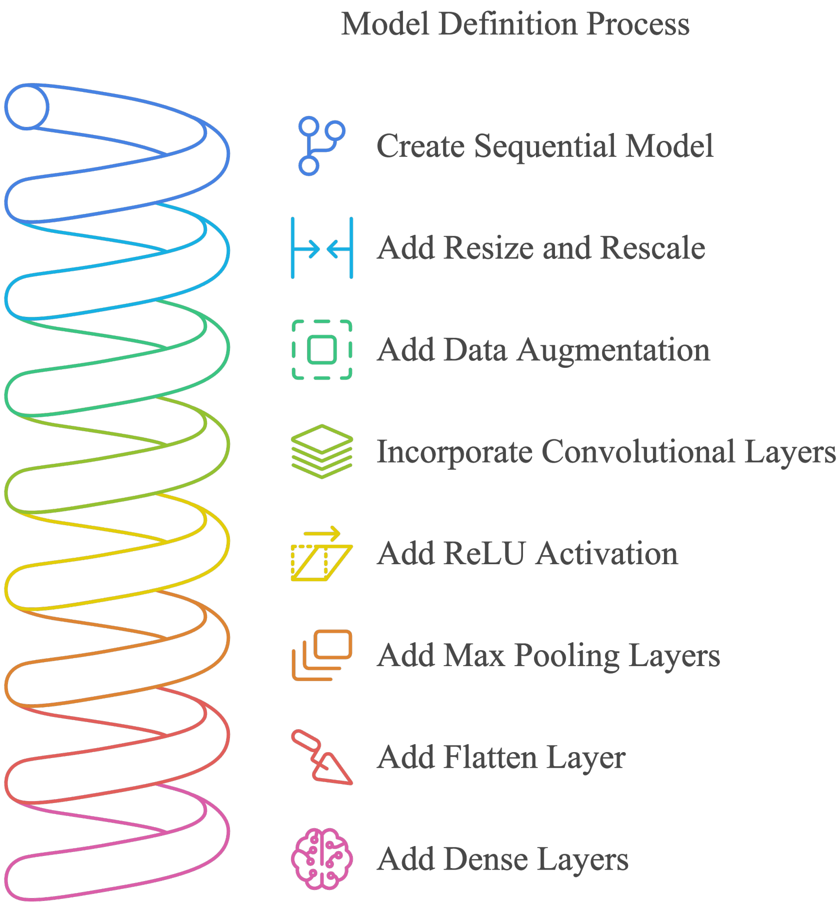


Figure 3.8: Model definition process

focusing on key design parameters.

3.2.4.2 Adding Resize and Rescale Layers

First, we start off with some preparatory components to add the resize and rescale layers to standardize the input images. Resizing would get the same dimensions for every input image, for example, 224×224 pixels, important to run within convolutional layers. Rescaling creates a pixel scale of [0, 1] enhancing numerical stability, hence speeding up the convergence with the training. This is an important step to ensure the uniformity of data fed into the model and to enable the latter to fetch features during further stages, highlighting how preprocessing is critical to the overall model performance.

3.2.4.3 Incorporating Data Augmentation Layers

Data augmentation increases the size of the training dataset by doing random transformations, such as rotation and flipping.These augmentations will ameliorate the predicament of model overfitting, acting to strengthen the ML model. They are vital for the preparatory steps taken to put the model in a position to generalize from its training, especially when infamously limited datasets are concerned.

3.2.4.4 Adding Convolutional Layers with ReLU and Max Pooling

The core of the model consists of convolutional layers that extract spatial features from the input images. Early layers capture simple patterns like edges, while deeper layers identify complex structures such as textures and shapes. The ReLU activation function introduces non-linearity, enabling the network to learn intricate patterns that linear models cannot. Max pooling layers follow the convolutional layers to downsample the feature maps, reducing spatial dimensions and computational complexity. This combination ensures efficient feature extraction and preserves the most important information, forming the foundation of the network's ability to understand image data.
The model involves many convolutional layers that derive spatial features from the input images. Early layers capture simple patterns such as edges, while deeper layers convey more complex structures such as textures and shapes. Slightly different from the other activation functions, the ReLU introduces non-linearity into the network, so that the learning process enables the learning of a more complex pattern than that which a linear regression could capture. Then come the max pooling layers, which downsample the feature maps to reduce their spatial dimensions and the complexity of following convolutional layers. This combination allows not only efficient feature retrieval but also retention of the most significant information, thus representing the basis on which the network can understand image data.

3.2.4.5 Flattening and Dense Layers

A flattening layer then converts the multi-dimensional outputs of the convolutional layers into one-dimensional vector inputs that can be fed into the dense layers. Dense layers implement full connections to learn high-level representations of the features learned from the

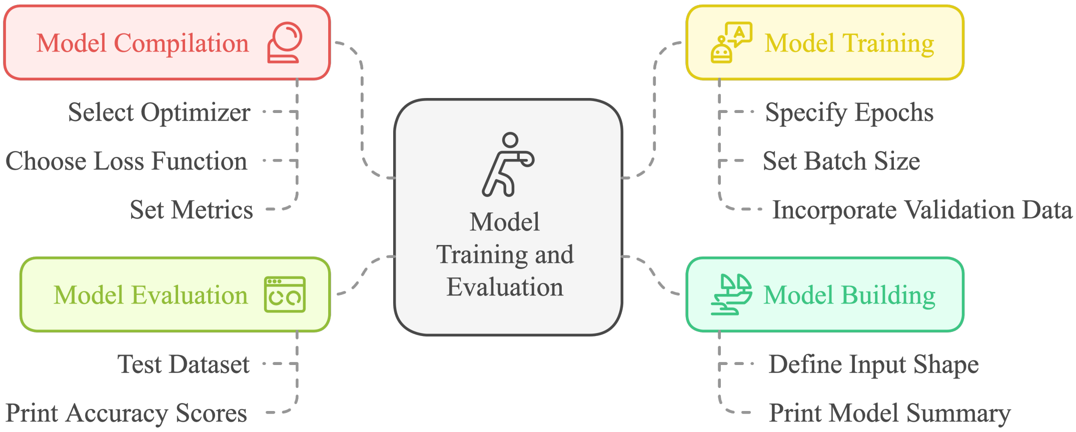


Figure 3.9: model training & Evaluation

convolutional layers. Intermediate dense layers operate with ReLU activation for non-linear learning while the last one employs a softmax activation function to give a probability distribution over classes. This architecture thus nicely interlinks the feature extraction with classification that creates a perfect fit in prediction towards the problem statement.

3.2.5 Model Training and Evaluation

The training and evaluation focuses on the building and optimizing of convolutional neural networks to automate scorecard analysis. The architecture uses input shape (256, 256, 3), which is customized for resized and rescaled scorecard images. The model is compiled with the Adamax optimizer to guarantee stable adaptive learning along with Sparse Categorical Cross-Entropy classification as the loss, which is ideal for multi-class classification of scorecard elements. Evaluation Metrics are established as the Model's Accuracy. Training occurs for 50 epochs, with a fixed batch size of 32 images per iteration, which solves the trade-off between computational efficiency and performance. The summary for this model thus validates its architecture, and tests performed on the testing dataset show that it was able to identify elements such as player names, scores, and balls faced accurately, therefore meeting the objective of automation of cricket scorecards.

3.2.5.1 Building the Model with Defined Input Shape

The proposed model is defined with the determined input shape, in terms of the height, the width, and the channels in the input images. For example, if the images are in RGB with dimensions of (224, 224), then the input shape is (224, 224, 3). When specifying the input dataset, one must ensure that it is in line with what the first layer of the neural network was designed to expect. It should be noted that specifying the input shape helps initialize the model and preallocate memory optimally, allowing modeling together for training purposes. For a thesis, it's mainly important because different input shapes would have an impact on the architecture and thus on compatibility with real-world input datasets.

3.2.5.2 Printing the Model Summary

The model summary presents a detailed overview of the network's architecture, specifically the type of each layer, output dimensions, and number of parameters. Thus it allows for the validation of a model design before training is even begun. For example, the summary provides insight into how the convolutional and pooling layers operate on the feature maps or how the dense layers map the collected features into class probabilities. An author presents model summary in a dissertation as a means of transparency to provide an understandable view of the architecture to the reading audience.

3.2.5.3 Compiling the Model

Compiling the model is an important step where critical training configurations are defined. The Adam optimizer has constructed its reputation by having an adaptive learning rate and by its fast, computationally efficient nature which allows for its greater use in complex models. The multi-class classification tasks may apply the sparse categorical cross-entropy loss with its integer-encoded class labels. It measures what optimization in predicting probability of the true label and guides the optimization. Accuracy metric, determined while compiling, keeps track of the performance of the model when it is being trained and, when it has been tested. In a dissertation, this phase endorses how the choice of optimizer and loss function meets the requirements of the problem and invokes sustainably advantageous learning tasks.

3.2.6 Training the Model

The model minimizes the loss function during training through weight adjustments by back-propagation. Training expects delivery of the training dataset through affirmation of a batch size that ensures a compromise between memory usage and computations. Epochs indicate how many times the dataset will pass through the model for each set of training data. Validation data show the model's ability to generalize to unseen data to avoid overfitting. From a thesis perspective, there should be a justification of the training parameters like batch size and epochs, and the comparison of training versus validation accuracy should aid in asserting the model's effectiveness.

3.2.6.1 Evaluating the Model on the Testing Dataset

The model is assessed for its applicability in the real world after being trained on the training dataset using the various testing datasets. Keeping accuracy as a yardstick provides a quantitative measurement of how well the model does in classifying unseen data. This is quite crucial as it directly reflects how much the model is used practically, validating its capability of generalization. The presentation of the evaluation results, like accuracy scores and maybe confusion matrices, gives empirical evidence of the robustness and efficiency of the model.

3.2.7 Prediction and Visualization

3.2.7.1 Extracting a Batch of Images and Labels from the Testing Dataset

The first step in this process is the extraction of a batch of images and their related labels from the test dataset, which consists of previously unseen data and is used strictly for evaluating the model's performance. A batch is a grouping of a certain number of images that are processed simultaneously, and choosing one of them thus guarantees computational efficiency. From a thesis standpoint, this step is crucial in demonstrating the performance of the model on individual examples and informing about its generalization capabilities.

3.2.7.2 Retrieving the First Image and Label

Once the batch is obtained, the first image along with its label is independently considered. This serves as an example in assessing the model's predictions. That label serves as the true ground truth and serves as a standard for comparison with what the model has outputted. Putting this example in a thesis allows the viewer to visualize the input and its label, which can concretely demonstrate the data's structure and the interpretability of the model.

3.2.7.3 Printing the Actual Label Based on Class Names

Classically, one maps a such label to its class name (the report illustrates using a few examples like "cat," "dog," etc.) using a certain predefined mapping between the numbers assigned with numerical indices and class names. It will definitely clear up the matter and help measure the model's comprehension of the data. It is significant to show an actual label in a thesis to confirm the predicted output's correctness with necessary evidence of the dataset's labeling consistency.

3.2.7.4 Predicting the Label Using the Model

An isolated image is given to the preceding model for the predicted label. The output from the model is the probability across all the classes, where the class with the highest probability receives the predicted output. This shows practical utilization of the model itself on the never-before-seen data. It is worth mentioning in the thesis that predictions give the room for building an evidence base for arguing how the model reasoned, plus how well its gained knowledge is generalized.

3.2.7.5 Printing the Predicted Label

This class, in turn, shall take its name according to the initial prediction and print it. This is the most important phase of comparison between actual label and predicted label; the result thus shows whether the model has come up with a right or wrong prediction. This comparison gives further insight into how well the model is performing at per sample, which, in turn, opens wine-drenched avenues to talk about enhancing model performance or conducting error analysis.

3.2.8 Plotting Training and Validation Accuracy Curves

The training history plots accuracy and loss curves during training and validation. These graphs show how the model's performance changes with respect to epochs regarding training and validation data.

- **Training accuracy curves:** How well the model learns from the training data.
- **Validation accuracy curves:** How well the model generalizes to unseen data.

Chapter 4

Results

Chapter 4

Results

4.1 Hyper Parameter Tuning

To Achieve the actual results; various parameters like Optimizer, number of Dense Layers, and Learning rates were changed updated during the time of experiments. Below are given some hyper parameter tuning setups:

4.1.1 Hyper Parameter Tuning Phase - I

In first phase of experiment, the architecture of the model was configured with five dense layers, each layer receiving variable input sizes. The RMSprop optimizer was used, with the learning rate fixed at 0.001 for all of the trials in this phase. The detailed configuration for this setup is given in Table 4.1, whereas the graphical representation of the model's performance for the same is illustrated in Figure 4.1.

Dense Layer	input of dense layers	Output of dense layers	Learning Rate	Accuracy
1	32	256	0.001	34.38
2	64	256	0.001	31.25
3	64	256	0.001	34.38
4	128	256	0.001	56.25
5	256	256	0.001	46.88

Table 4.1: Hyper Parameter Tuning - 1

Figure 4.2 displays the low accuracy results for the trained model. The umpire doing gesture of "Four" was given to test the accurracy of the proposed model; but it predicted "six" as 39%, "Out" as 32%, and "Four" as only 28%; which showcases that this is not feasible for prediction in real time environment.

4.1.2 Hyper Parameter Tuning Phase - II

In the second phase, the number of dense layers used were set to 6, the number of dense layer inputs were set to 6 with different inputs of dense layers. The RMS Prop was used as optimizer and the learning rates for the experiments were set to 0.001. Table 4.2 shows

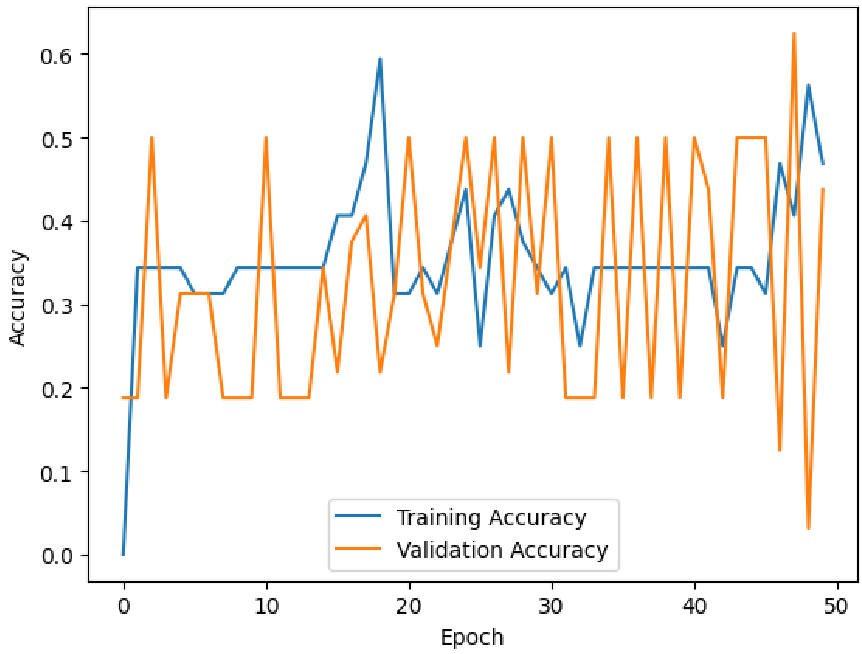


Figure 4.1: Hyper Parameter Tuning - 1 Graph

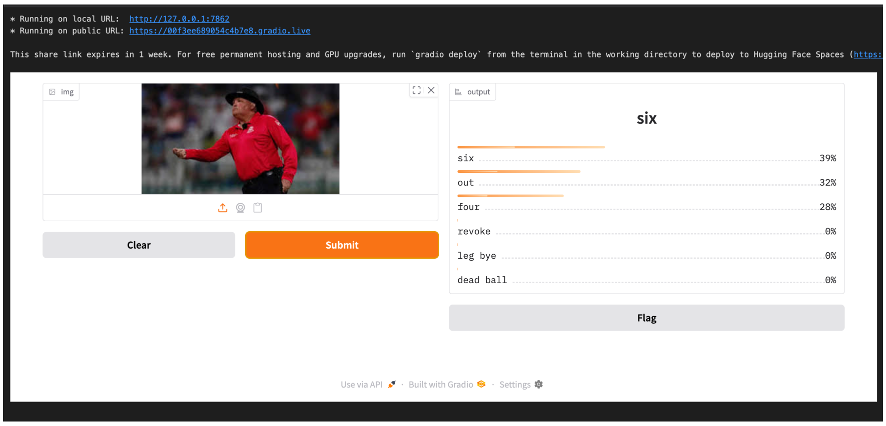


Figure 4.2: Hyper Parameter Tuning 1 Experimental Result

the detailed configurations of first experimental setup, while Figure 4.3 shows the graphical representation of the results gained.

Figure 4.4 displays the low accuracy results for the trained model. The umpire doing gesture of “Four” was given to test the accurracy of the proposed model; but it predicted “out” as 38%, “six” as 24%, and “Four” as only 38%; which showcases that this is not feasible for prediction in real time environment.

Dense Layer	input of dense layers	Output of dense layers	Learning Rate	Accuracy
1	32	256	0.001	56.25
2	32	256	0.001	56.25
3	64	256	0.001	56.25
4	64	256	0.001	56.25
5	128	256	0.001	56.25
6	256	256	0.001	59.38

Table 4.2: Hyper Parameter Tuning - 2

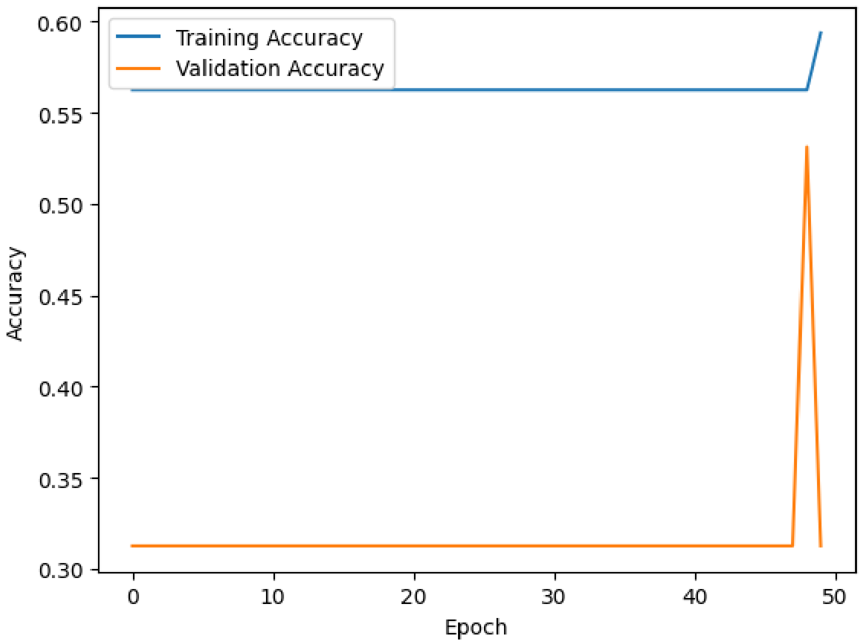


Figure 4.3: Hyper Parameter Tuning - 2 Graph

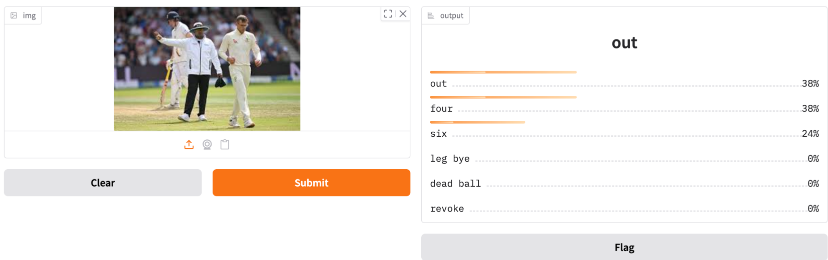


Figure 4.4: Hyper Parameter Tuning 2 experimental result

Dense Layer	input of dense layers	Output of dense layers	Learning Rate	Accuracy
1	32	256	0.001	34.38
2	64	256	0.001	37.38
3	64	256	0.001	40.62
4	128	256	0.001	43.75
5	128	256	0.001	43.75
6	256	256	0.001	43.75

Table 4.3: Hyper Parameter Tuning - 3

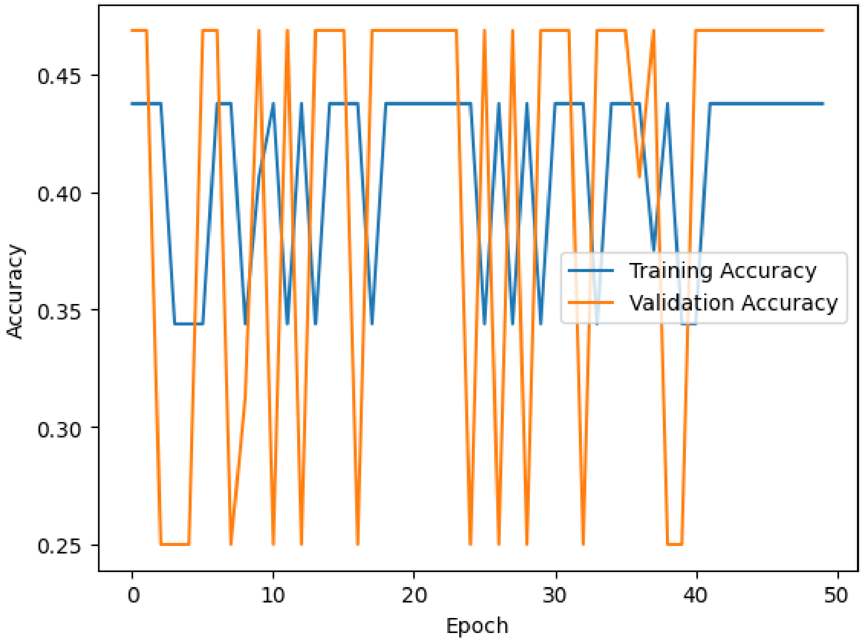


Figure 4.5: Hyper Parameter Tuning - 3 Graph

4.1.3 Hyper Parameter Tuning Phase - III

In the Third phase, the number of dense layers used were set to 6, the number of dense layer inputs were set to 6 with different inputs of dense layers. The RMS Prop was used as optimizer and the learning rates for the experiments were set to 0.001. Table 4.3 shows the detailed configurations of first experimental setup, while Figure 4.5 shows the graphical representation of the results gained.

Figure 4.6 displays the low accuracy results for the trained model. The umpire doing gesture of “Four” was given to test the accurracy of the proposed model; but it predicted “six” as 50%, “Out” as 37%, and “Four” as only 13%; which showcases that this is not feasible for prediction in real time environment.

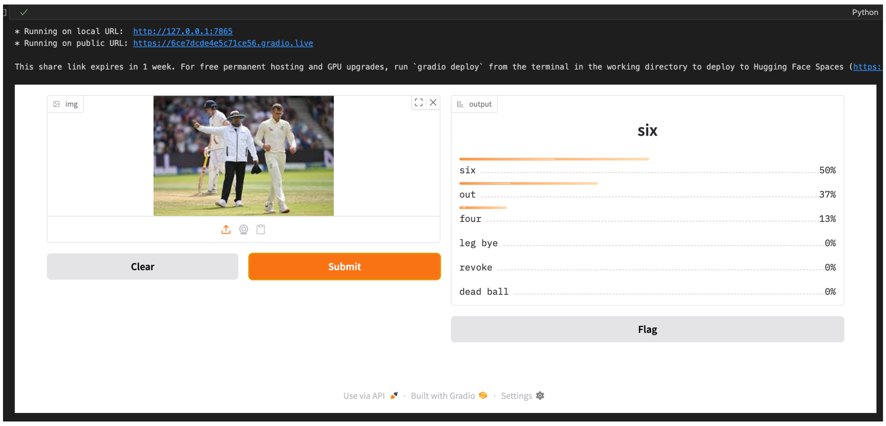


Figure 4.6: Hyper Parameter Tuning 3 Experimental Result

4.1.4 Hyper Parameter Tuning Phase - IV

In the Fourth phase, the number of dense layers used were set to 6, the number of dense layer inputs were set to 6 with different inputs of dense layers. The Adam optimizer was used as optimizer and the learning rates for the experiments were set to 0.001. Table 4.4 shows the detailed configurations of first experimental setup, while Figure 4.7 shows the graphical representation of the results gained.

Dense Layer	input of dense layers	Output of dense layers	Learning Rate	Accuracy
1	32	128	0.001	28.12
2	32	128	0.001	34.38
3	64	128	0.001	31.28
4	64	128	0.001	34.38
5	128	128	0.001	50.00
6	128	128	0.001	43.75

Table 4.4: Hyper Parameter Tuning - 4

Figure 4.10 displays the low accuracy results for the trained model. The umpire doing gesture of “Four” was given to test the accurracy of the proposed model; but it predicted “four” as 47%, which is a correct prediction but with low accuracy while also predicting “six” as 42%, and “out” as only 12%; which showcases that this is not feasible for prediction in real time environment.

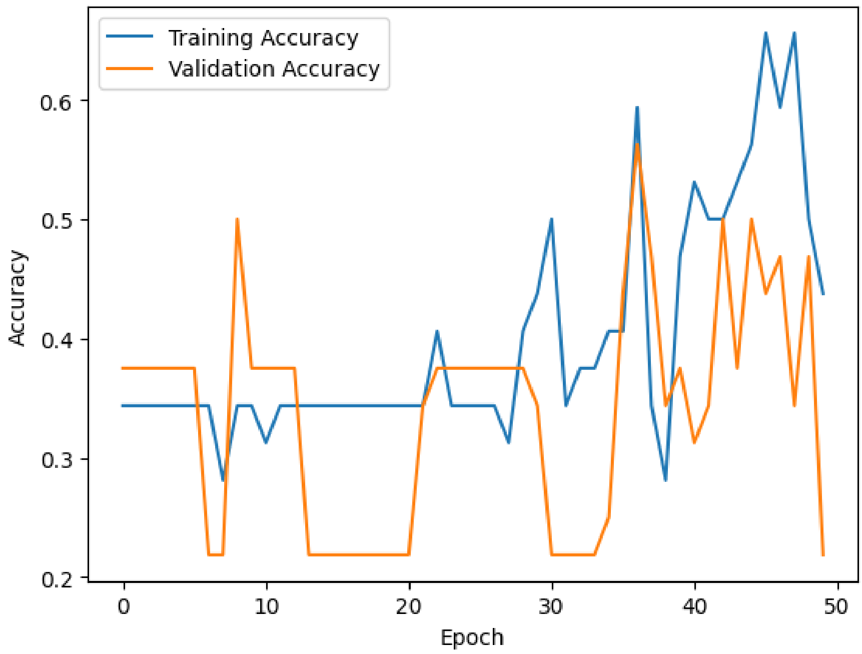


Figure 4.7: Hyper Parameter Tuning - 4 Graph

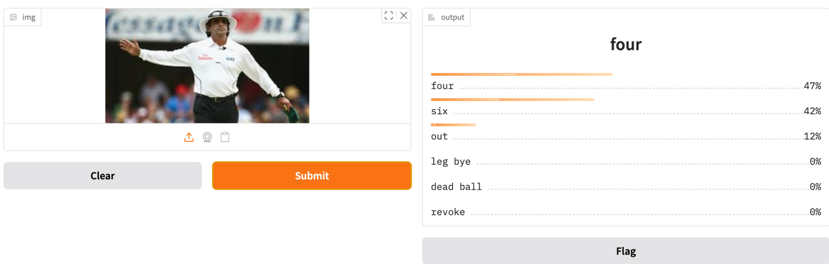


Figure 4.8: Hyper Parameter Tuning 4 Experimental Result

4.1.5 Hyper Parameter Tuning Phase - V

In the Fifth phase, the number of dense layers used were set to 5, the number of dense layer inputs were set to 5 with different inputs of dense layers. The Adam optimizer was used as optimizer and the learning rates for the experiments were set to 0.001. Table 4.5 shows the detailed configurations of first experimental setup, while Figure 4.11 shows the graphical representation of the results gained.

Figure **??** displays the low accuracy results for the trained model. The umpire doing gesture of “Four” was given to test the accurracy of the proposed model; but it predicted “six” as 42%, “Out” as 12%, and “Four” as only 47%; which showcases that this is not

Dense Layer	input of dense layers	Output of dense layers	Learning Rate	Accuracy
1	32	64	0.01	40.62
2	32	64	0.01	31.25
3	64	64	0.01	37.50
4	64	64	0.01	65.62
5	64	64	0.01	71.88

Table 4.5: Hyper Parameter Tuning - 5

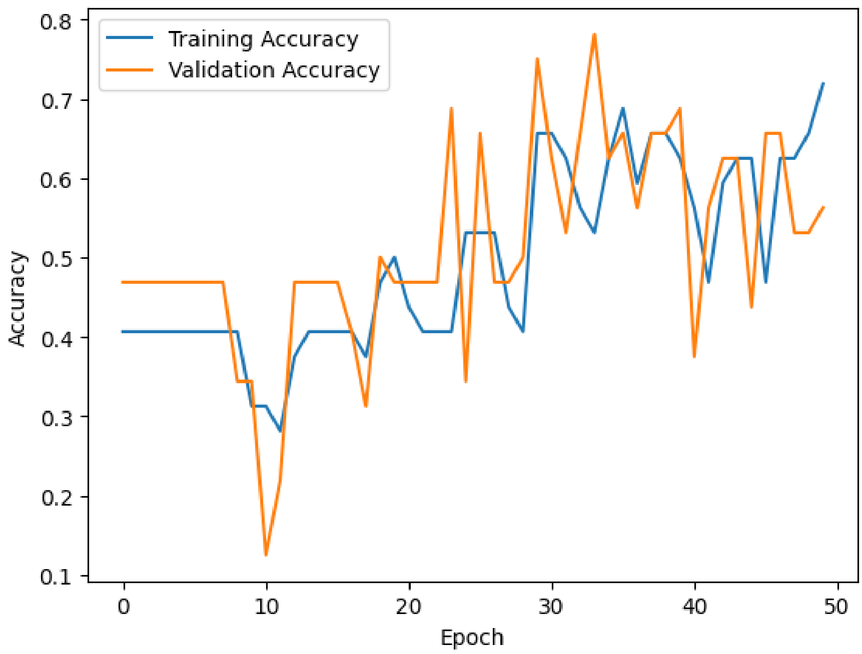


Figure 4.9: Hyper Parameter Tuning - 5 Graph

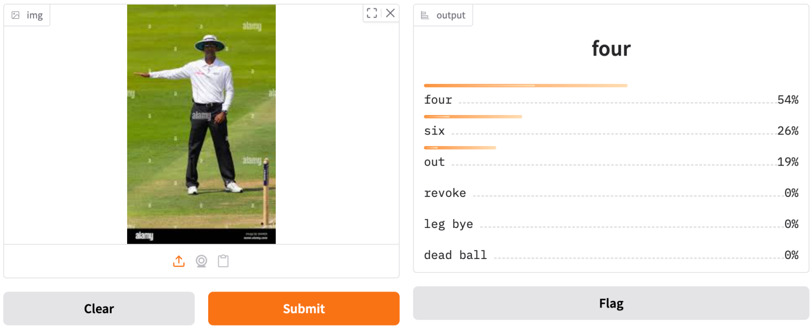


Figure 4.10: Hyper Parameter Tuning 4 Experimental Result

feasible for prediction in real time environment.

4.1.6 Hyper Parameter Tuning Phase - VI

In the Sixth phase, the number of dense layers used were set to five, the number of dense layer inputs were set to 6 with different inputs of dense layers. The Adam optimizer was used as optimizer and the learning rates for the experiments were set to 0.001. Table 4.6 shows the detailed configurations of first experimental setup, while Figure **??** shows the graphical representation of the results gained.

Dense Layer	input of dense layers	Output of dense layers	Learning Rate	Accuracy
1	64	256	0.01	37.52
2	64	256	0.01	65.62
3	128	256	0.01	70.61
4	128	256	0.01	82.98
5	256	256	0.01	88.92
6	256	256	0.01	98.27

Table 4.6: Hyper Parameter Tuning - 6

Figure 4.12 displays the low accuracy results for the trained model. The umpire doing gesture of "Six" was given to test the accurracy of the proposed model; but it predicted "six" as 98%, "Out" as 02%, which showcases that this approach can be suitable for the real time implementation purposes.

Based on the results shown in the above given test cases (Hyperparameter Tuning I - VI), It is clear that the Adam optimizer when configured with six layers, and provided with the input and output parameters as given in Table 4.6, gives the most accurate model for score-

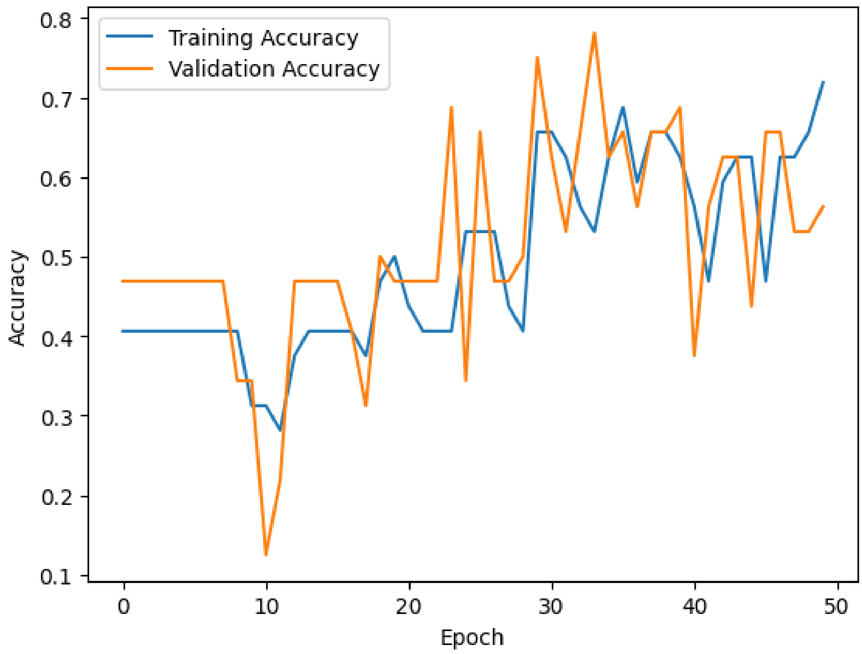


Figure 4.11: Hyper Parameter Tuning - 5 Graph

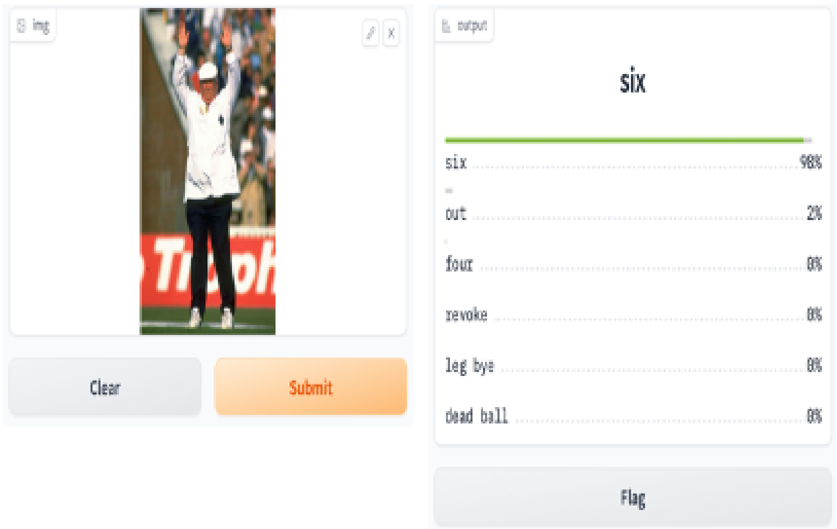


Figure 4.12: Hyper Parameter Tuning 6 Experimental Result

card prediction. The detailed analysis of these given results are provided in the upcoming sections of this chapter.

4.2 Results presented by the model

In Figure 4.13, the graph demonstrates the performance metrics of the model as evaluated on a validation process. Figure 4.13 would depict the training accuracy and validation accuracy (val_accuracy) curves over the number of epochs in the training of the model. The curves represent the working of the model in predicting correct labels for training data (Training accuracy) against unseen data (val_accuracy), in pursuit of their maximization-signaling high generalization of the model.

Figure 4.13: validation and training accuracy test

4.2.1 Comparison of different methods with proposed model

The performance of four different predictive models, namely Logistic Regression, LSTM, Inception V3, and Proposd CNN, is evaluated using important metrics such as accuracy, precision, recall, and F1 score. The results of this comparison are presented in Table 4.7, and Figure 4.14, respectively. The Logistic Regression (LR) model demonstrates a great overall performance, with an accuracy of 91.06% and precision and recall that are closely aligned at approximately 88%. This indicates that the model is capable of making accurate predictions. LSTM is slightly behind in accuracy, coming in at 90%, but it maintains consistent performance across all metrics, coming in at 89%. This demonstrates that it is effective at

Table 4.7: Comparison between existing technologies and proposed work

Model	Accuracy	Precision	Recall	F1-Score
logistic Regression [12]	91.06%	88.10%	88.13%	88.13%
Long Short-Term Memory [46]	90%	89%	89%	89%
Inception V3 [47]	66.10%	75%	75%	75%
Proposed CNN	98.25%	97.23%	95.60%	97.10%

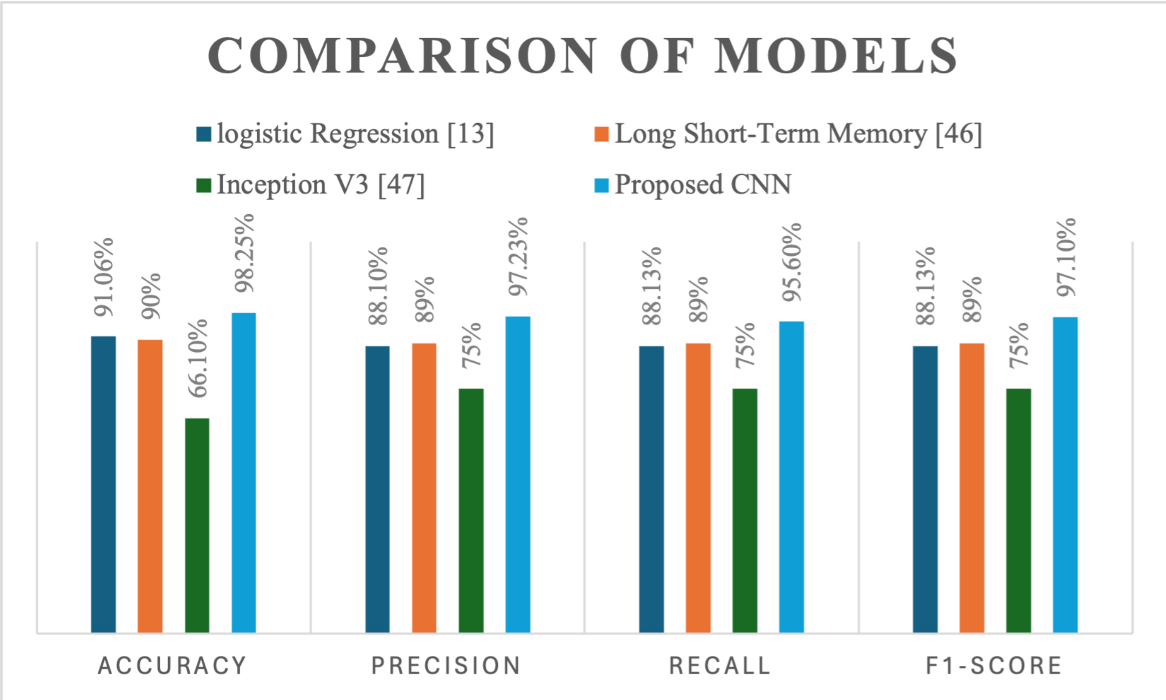
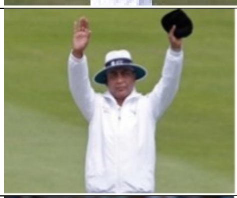


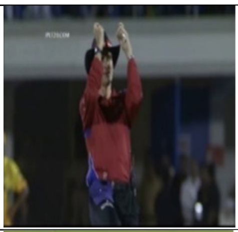
Figure 4.14: Comparison between existing technologies and proposed work

balancing prediction that is true positive and prediction that is true negative. The fact that Inception V3 has the lowest accuracy, which is 66.10%, and the uniform metrics, which are 75%, indicates that it has a greater amount of difficulty accurately classifying situations in comparison to the other versions. CNN shines as the top performance with the highest accuracy at **98.25%**, superior precision, and an F1-score that is **above 97%**. However, its recall at **95.60%** demonstrates that it somewhat trails behind in identifying all relevant cases. The purpose of this overview is to emphasise the merits and limits of each model, with a particular emphasis on CNN's remarkable capacity to make accurate predictions with minimum errors, and to advise that models be selected through careful consideration.

4.2.2 Accuracy Testing & Calculation

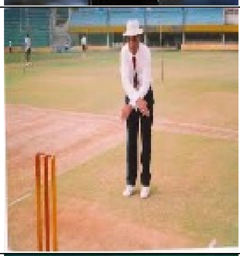
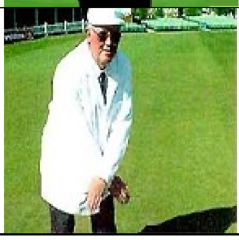
Table 4.8: Accuracy Test

Umpire Signal Image	Expected Output	Proposed Model Output	Prediction Status
	Six	Six	Correct
	Six	Six	Correct
	Six	Six	Correct
	Six	Six	Correct
	Six	Six	Correct

Umpire Signal Image	Expected Output	Proposed Model Output	Prediction Status
	Six	Out	Incorrect
	Out	Out	Correct
	Out	Out	Correct
	Out	Out	Correct
	Out	Out	Correct
	Out	Out	Correct
	Four	Four	Correct

Umpire Signal Image	**Expected Output**	**Proposed Model Output**	**Prediction Status**
	Four	Four	Correct
	Four	Four	Correct
	Four	Four	Correct
	Four	Four	Correct
	Revoke	Revoke	Correct
	Revoke	Revoke	Correct
	Revoke	Revoke	Correct

Umpire Signal Image	Expected Output	Proposed Model Output	Prediction Status
	Revoke	Revoke	Correct
	Revoke	Revoke	Correct
	Leg Bye	Leg Bye	Correct
	Leg Bye	Leg Bye	Correct
	Leg Bye	Leg Bye	Correct
	Leg Bye	Leg Bye	Correct
	Leg Bye	Leg Bye	Correct

Umpire Signal Image	Expected Output	Proposed Model Output	Prediction Status
	Dead Ball	Dead Ball	Correct
	Dead Ball	Dead Ball	Correct
	Dead Ball	Dead Ball	Correct
	Dead Ball	Dead Ball	Correct
	Dead Ball	Dead Ball	Correct

The accurate calculation of average accuracy can be done as follows:

- For every gesture G_i, where i varies from 1 to the entire count of distinct gestures, the gained accuracy for the Umpire Singal is:

$$\text{Accuracy}_{G_i} = \frac{\sum_{j=1}^{n_i} \text{Correct}_j}{n_i}$$

- Here, n_i is the aggregate count for the predictions of signal G_i.

- The Value of $Correct_j$ is equals to 1 only if the j^{th} estimate for the given signal (gesture) G_i is accurate, and 0 if it is not.

As Shown in Figure 4.15, Figure 4.16; the overall calculated accuracy for the proposed model is:

To calculate the overall average accuracy, aggregate the accuracies for each gesture and divide by the total number of distinct gestures

$$\text{Overall Average Accuracy} = \frac{\sum_{i=1}^{N} \text{Accuracy}_{G_i}}{N}$$

Here:

- N is the total number of gestures shown by Umpire.

Calculations as per tests shown in Figure 4.15, Figure 4.16, the manual tests:

$$\text{Overall Average Accuracy} = \frac{0.833 + 1 + 1 + 1 + 1 + 1}{6} = \frac{5.833}{6} \approx 0.9722$$

The manual evaluation of the presented convolutional neural network (CNN) algorithm indicates an overall average accuracy of roughly 97.22% across all gestures.

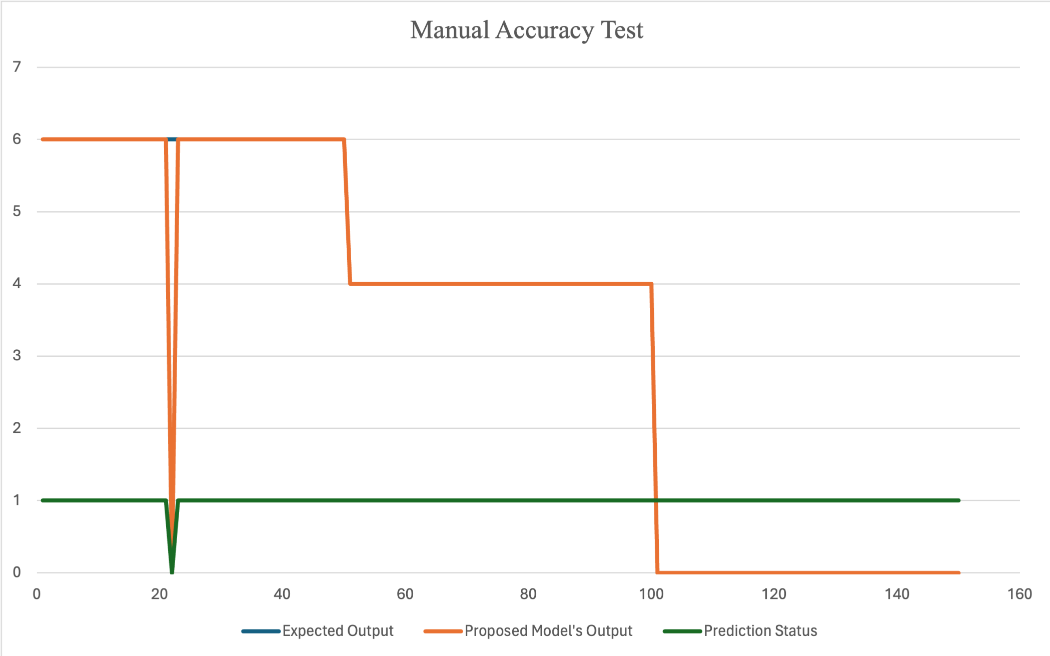


Figure 4.15: Manual Accuracy Test - 1

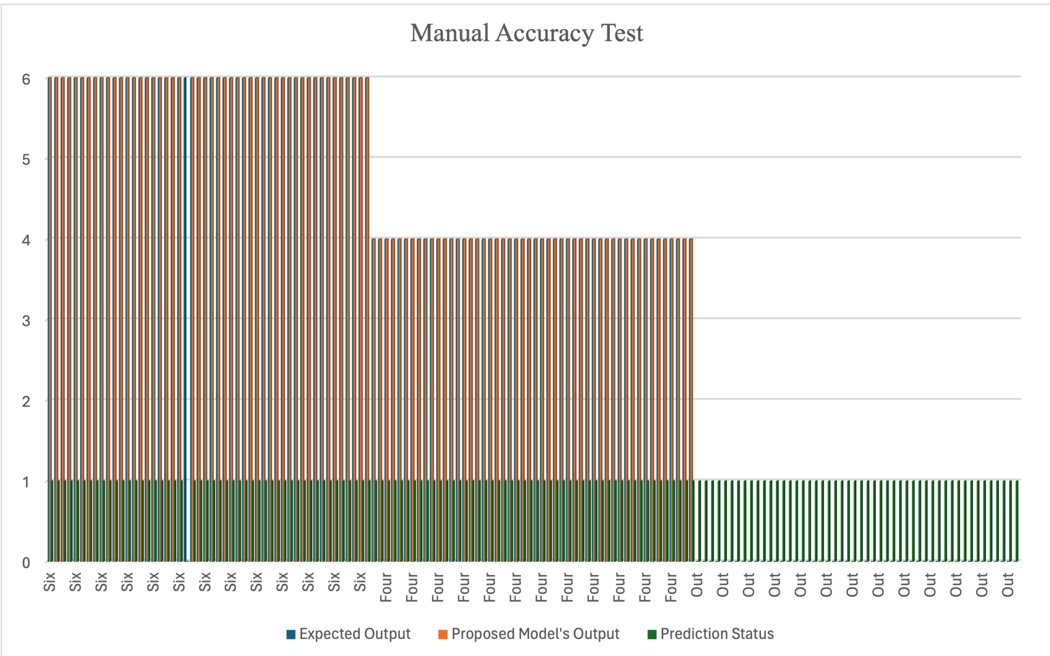


Figure 4.16: Manual Accuracy Test - 2

Chapter 5

Conclusion & Future Work

Chapter 5

Conclusion & Future Work

5.1 Conclusion

The presented study shows that how the Convolutional Neural Network (CNN) model can analyze and classify the different - different parts of games (specifically for cricket game in this research). The average accuracy gained from the proposed work is 98%, which beats all of the other methods tested till date. This makes a big stride over the current ways to sort the actions of game, and scoring in it. The proposed work mentions how well the model works for sports data analysis comparing it to old methods. The CNN model is more accurate and precise than these older approaches.

5.2 Future Work

New neural network designs bigger datasets with more game types, and live data analysis are all exciting ideas. The mix of different fields and the need to think about ethics when using AI for sports data show this research could go in many directions. This points to a promising future where AI and sports come together.

References

[1] K. Bharath, “Introduction to Deep Neural Networks — DataCamp,” 7 2023.

[2] “An introduction to machine learning, its importance, types, and applications.”

[3] “A Comprehensive Guide to Convolutional Neural Networks - Shiksha Online.”

[4] “What is Deep Learning How is It Redesigning the Tech World.”

[5] V. Vec, S. Tomažič, A. Kos, and A. Umek, “Trends in real-time artificial intelligence methods in sports: a systematic review,” *Journal of Big Data*, vol. 11, 10 2024.

[6] D. V. Dey, “The Role of Artificial Intelligence in Physical Education and Sports: A Review of Current Applications and Future Potential,” *JOURNAL GLOBAL VALUES*, vol. XIV, pp. 9–14, 12 2023.

[7] R. Mitra, *3D Modelling from Images using Semi Supervised Approaches*. PhD thesis, Indian Institute of Technology Bombay, Mumbai, 2023.

[8] G. Wagner, R. Lukyanenko, and G. Paré, “Artificial intelligence and the conduct of literature reviews,” *Journal of Information Technology*, vol. 37, no. 2, pp. 209–226, 2022.

[9] S. Dc, “Artificial intelligence in sport: an ethical issue,” *Unity journal*, vol. 3, pp. 27–39, 1 2022.

[10] K. Kousha and M. Thelwall, “Artificial intelligence technologies to support research assessment: A review,” *arXiv preprint arXiv:2212.06574*, 6 2022.

[11] A. Kilani, A. B. Hamida, and H. Hamam, *Artificial intelligence review*. IGI Global, Encyclopedia of Information Science and Technology, fourth ed., 1 2018.

[12] M. A. Shahjalal, Z. Ahmad, R. Rayan, and L. Alam, "An approach to automate the scorecard in cricket with computer vision and machine learning," in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, pp. 1–6, 12 2017.

[13] "(596) icc men's t20 world cup 2024 - youtube," 2 2024.

[14] H. Shingrakhia and H. Patel, "SGRNN-AM and HRF-DBN: a hybrid machine learning model for cricket video summarization," *The Visual Computer*, vol. 38, no. 7, pp. 2285–2301, 2022.

[15] A. Saha, N. Kothari, P. Shah, R. Parekh, and N. Shekokar, "Cricket Highlight Generation: Automatic Generation Framework Comprising Score Extraction and Action Recognition," in *2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT)*, pp. 1–7, 10 2022.

[16] Vidisha and V. Bhatia, "A review of Machine Learning based Recommendation approaches for cricket," in *2020 Sixth International Conference on Parallel, Distributed and Grid Computing (PDGC)*, pp. 421–427, 11 2020.

[17] D. J. Vestly, S. Hariharan, V. Kukreja, A. B. Prasad, K. Swaraj, and D. Gopichand, "Parametric Analysis of a Cricketer's Performance using Machine Learning Approach," in *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 344–348, 5 2023.

[18] M. Wyawahare, A. Dhanawade, S. Dharyekar, A. Dhole, and M. Dhopade, "Automating Scorecard and Commentary Based on Umpire Gesture Recognition," in *Advancements in Smart Computing and Information Security* (S. Rajagopal, P. Faruki, and K. Popat, eds.), (Rajkot, India), pp. 158–168, Springer Nature Switzerland, 1 2023.

[19] J. Kumar, N. Prince, and H. K. Baker, "Balanced Scorecard: a systematic literature review and future research issues," *FIIB Business Review*, vol. 11, pp. 147–161, 1 2021.

[20] K. H. Thanoon, A. F. Shareef, and O. A. Alsaif, "Digital Processing and Deep Learning Techniques: A review of the Literature," *NTU Journal of Engineering and Technology*, vol. 1, 1 2022.

[21] Z. I. Azhari, S. Setumin, A. D. Rosli, and S. J. A. Bakar, “A systematic literature review on hardware implementation of image processing,” *International Journal of Reconfigurable and Embedded Systems*, vol. 12, pp. 19–28, 3 2023.

[22] S. Kumar, W. M. Lim, R. Sureka, C. J. C. Jabbour, and U. Bamel, “Balanced scorecard: trends, developments, and future directions,” 8 2024.

[23] Y. Wang, G. Shan, H. Li, R. Feng, Y. Zhang, G. Li, and L. Wang, “A Deep-Learning-Based Method for Gaining More Biomechanical Parameters with Fewer Sensors in Fast and Complex Movements,” in *2022 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 704–709, 2022.

[24] T. Lee, V. P. Singh, and K. H. Cho, “Tensorflow and Keras Programming for Deep Learning,” in *Deep Learning for Hydrometeorology and Environmental Science*, vol. 99, pp. 151–162, Cham: Springer Nature Switzerland AG, 1 2021.

[25] University of Central Florida, “Ucf101 - action recognition data set.”

[26] W. Ahmad, M. Munsif, H. Ullah, M. Ullah, A. A. Alsuwailem, A. K. Jilani Saudagar, K. Muhammad, and M. Sajjad, “Optimized deep learning-based cricket activity focused network and medium scale benchmark,” *Alexandria Engineering Journal*, vol. 73, pp. 771–779, 7 2023.

[27] “Mamdani and Sugeno Fuzzy inference Systems.”

[28] M. I. N. Faizal, A. H. A. Hassan, and F. D. Davaraj, “Development of a three-wheel cricket bowling machine integrated with fuzzy logic controller,” in *2018 IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE)*, pp. 7–12, 2018.

[29] S. Sakib, M. A. Haque Mridha, N. Hasan, M. T. Habib, and M. S. Mahbub, “Event Detection from Cricket Videos Using Video-Based CNN Classification of Umpire Signals,” in *2022 4th International Conference on Sustainable Technologies for Industry 4.0 (STI)*, pp. 1–5, Institute of Electrical and Electronics Engineers Inc., 12 2022.

[30] “3D Pose Detection with MediaPipe BlazePose GHUM and TensorFlow.js.”

[31] M. Devanandan, V. Rasaratnam, M. K. Anbalagan, N. Asokan, R. Panchendrarajan, and J. Tharmaseelan, "Cricket Shot Image Classification Using Random Forest," in *2021 3rd International Conference on Advancements in Computing (ICAC)*, (Virtual Mode), pp. 425–430, IEEE, 12 2021.

[32] M. U. Haq, M. A. J. Sethi, S. Ahmad, M. A. Elaffendi, and M. Asim, "Automatic Player Face Detection and Recognition for Players in Cricket Games," *IEEE Access*, vol. 12, pp. 41219–41233, 2024.

[33] Y. Zhou, "Recognition for Hybrid Modulation Based on Convolutional Neural Network," in *2023 International Conference on Image Processing, Computer Vision and Machine Learning (ICICML)*, pp. 764–768, 2023.

[34] H. Song, C. Wang, K. Shen, C. Geng, B. Yang, and Y. Wang, "Research of Gesture Recognition Based on Improved Convolutional Neural Network and Support Vector Machine," in *2022 China Automation Congress (CAC)*, pp. 1232–1237, 11 2022.

[35] R. Chaudhary, S. Bhardwaj, and S. Lakra, "A DEA Model for Selection of Indian Cricket Team Players," in *2019 Amity International Conference on Artificial Intelligence (AICAI)*, pp. 224–227, 2019.

[36] M. M. Hatharasinghe and G. Poravi, "Data Mining and Machine Learning in Cricket Match Outcome Prediction: Missing Links," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pp. 1–4, 2019.

[37] R. Huang, H. Wang, F. Liu, and W. W. Guo, "Design of cricket automatic control system based on computer vision technology," in *Proceedings - 2019 International Conference on Robots and Intelligent System, ICRIS 2019*, pp. 26–29, Institute of Electrical and Electronics Engineers Inc., 6 2019.

[38] N. Batra, H. Gupta, N. Yadav, A. Gupta, and A. Yadav, "Implementation of augmented reality in cricket for ball tracking and automated decision making for no ball," *Proceedings of the 2014 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2014*, pp. 316–321, 11 2014.

[39] R. Kushwaha, G. Kaur, and M. Kumar, "Hand Gesture Based Sign Language Recognition Using Deep Learning," *ICSCCC 2023 - 3rd International Conference on Secure Cyber Computing and Communications*, pp. 293–297, 2023.

[40] G. N. Iyer, B. Vignesh, B. Sohan, R. Dharmesh, and V. Raman, "Automated Third Umpire Decision Making in Cricket Using Machine Learning Techniques," *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020*, pp. 1216–1221, 5 2020.

[41] K. R. Raval and M. M. Goyani, "A survey on event detection based video summarization for cricket," *Multimedia Tools and Applications*, vol. 81, pp. 29253–29281, 8 2022.

[42] A. Sen, K. Deb, P. K. Dhar, and T. Koshiba, "Cricshotclassify: An approach to classifying batting shots from cricket videos using a convolutional neural network and gated recurrent unit," *Sensors*, vol. 21, no. 8, 2021.

[43] M. F. A. Foysal, M. S. Islam, A. Karim, and N. Neehal, "Shot-Net: A Convolutional Neural Network for Classifying Different Cricket Shots," in *Communications in Computer and Information Science*, vol. 1035, 2019.

[44] "An Introduction to Computer Vision: Techniques & Real-Life Applications — by Quantum — Geek Culture — Medium."

[45] A. Mumuni and F. Mumuni, "Data augmentation: A comprehensive survey of modern approaches," *Array*, vol. 16, p. 100258, 12 2022.

[46] V. Kumar, M. Pruthi, A. Katyal, R. Semwal, and A. professor, "Cricket Gesture Detection Using LSTM, Media Pipe And CNN," *Journal of Xidian University*, vol. 7, 10 2023.

[47] A. Ravi, H. Venugopal, S. Paul, and H. R. Tizhoosh, "A Dataset and Preliminary Results for Umpire Pose Detection Using SVM Classification of Deep Features," *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, pp. 1396–1402, 7 2018.

Chapter A

List of Publications

List of Publications

[1] Jindal, V., Jindal, L., Bandhu, K. C., & Litoriya, R. (n.d.). *Automating cricket scorecards with deep Neural Network: A Next-Generation approach* (Vol. 2025). IET Conference Proceedings.

Abstract:
This work presents an approach to the automation of scorecards in cricket using CNN. The objective will be to develop such a system that will extract information that is relevant automatically and efficiently from actual cricket matches and provide the creation of a scorecard. A CNN model proposed in this study will be trained on a cricket match data set of considerable size, which will help to recognize and analyze different features of the game, which may consist of runs scored, wickets taken, and player performance. Automation in this area, in combination with this technique, could lead to a reduction in the amount of time and effort that cricket statisticians and broadcasters have to put in, while ensuring accurate, up-to-date scorecards with an accuracy of 95%, which is higher than currently available by 7%.
Keywords: Cricket Matches, Cricket Umpire Images, CNN, Machine Learning, Deep Neural Network.

Gmail

Vivek Jindal <vivek.jindal.sbg@gmail.com>

PiCET-2025: -Notification of Acceptance for Paper-ID 732 and Guidelines for Registration & Copyright form Submission

1 message

Microsoft CMT <email@msr-cmt.org> 1 April 2025 at 13:36
Reply-To: Swapnil M Parikh <swapnil.parikh17761@paruluniversity.ac.in>
To: Vivek Jindal <vivek.jindal.sbg@gmail.com>
Cc: picet@paruluniversity.ac.in

Dear Author(s)

On behalf of editorial board of PiCET-2025, we are happy to inform you that your paper has been ACCEPTED (based on the review reports) for presentation in PiCET-2025 conference in hybrid mode to be organized by Parul University, Vadodara during 02-03 May 2025.

Please refer to the reviewers' comments attached below which are intended to help you to improve your paper for final publication. You are advised to address the comments of the reviewers in your final paper suitably. Kindly note that the conference committee follows a rigorous double-blind peer-review system for screening and reviewing papers for the conference proceedings.

Instructions for Author(s) Registration & Payment of Registration Fee

----

(a) The details of registration fee as well as guidelines may be referred to at: https://picet.in/registration_fees.php

(b) Refer Camera Ready Paper Format: https://picet.in/assets/img/Manuscript_Preparation_Guidelines.doc

(c) Copyright Form can be downloaded from:https://picet.in/assets/img/IET%20Copyright%20form.pdf

(d) Kindly pay the conference registration fee via the following registration form and payment link(s):
https://easebuzz.in/link/GPJO0?

(e) After completion of payment through the above link--you are requested to fill up the form with all details: https://forms.gle/qfW8NoJK2txsnC9w7

*****EDITORIAL COMMENTS*****

Authors are advised to strictly incorporate the following points in the final paper:

1. The content matching must be within 15%. Single source matching not to exceed 2%.
2. All authors name, affiliation and mail ids must be there and the corresponding author to be marked.
3. No biography or photos of authors should be there.
4. Authors must check the English language carefully.
5. The work must be original and should not have been submitted or published in any other conf or journal. At any point of time, if it is being traced that the paper/content is AI-generated then the paper will be rejected without refund of registration fee.
6. Use uniform style for headings and subheading as per template.
7. The figures given in the papers should be 800dpi so that the picture clarity can be maintained.
8. All figures, equations and tables must be numbered properly and should be cited within the text.

9. Add missing publication details in references. All the references must be cited within the text.

******REVIEWER COMMENTS***********
Provide a more detailed explanation of the data analysis process.

Discuss the implications of the findings more thoroughly.

Regards,
Team PiCET 2025

Automating Cricket Scorecards with Deep Neural Network: A Next-Generation Approach

Vivek Jindal[1*], Latika Jindal[2], Kailash Chandra Bandhu[3], Ratnesh Litoriya[4]
vivek.jindal.sbg@gmail.com[1*], latika19mehrotra@gmail.com[2], kailashchandra.bandhu@gmail.com[3], litoriya.ratnesh@gmail.com[4]
[1*,2,3,4]Computer Science and Engineering, Medi-Caps University, Indore, India

Abstract: This work presents an approach to the automation of scorecards in cricket using CNN. The objective will be developing such a system that will extract information that is relevant automatically and efficiently from actual cricket matches and provide the creation of a scorecard. A CNN model proposed in this study will be trained on a cricket match data set of considerable size, which will help to recognize and analyze different features of the game, which may consist of runs scored, wickets taken, and player performance. Automation in this area, in combination with this technique, could lead to a reduction in the amount of time and effort that cricket statisticians and broadcasters have to put in, while ensuring accurate, up-to-date scorecards with an accuracy of 95%, which is higher than currently available by 7%.

Keywords: Cricket Matches, Cricket Umpire Images, CNN, Machine Learning, Deep Neural Network.

1. Introduction

While ensuring the inclusion of umpire signaling in the proposed methodology, the approach for the scorecard automation on cricket is believed to be a momentum-filled partnership between convolutional neural networks (CNN) and machine learning, all with regard to cricket matches held since 2018. The first phase is a structured technique involving systematic data collection on player details, match statistics, and umpire signals, for a number of cricket matches so far. An extensive data set is collated and rigorously pre-processed in terms of uniformity and reliability. Feature engineering prepares a structured input for the CNN architecture that underlies. The architecture possesses several pooling layers and convolutional layers so as to extract spatial features of the images relating to umpire signals and match details. This allows for an in-depth interpretation of umpire gestures and match dynamics at play. Simultaneously, a machine learning model is introduced to scrutinize the extracted features in order to predict umpire decisions and allow for automatic creation of the scorecards. The model is trained using historical data while tuning its parameters to ensure optimal performance [1].

Validation and testing measure the precision and efficiency of the system, using a different test set for robustness and generalizability. The flowchart describes processes like automated scorecard generation and umpire signaling interpretation that work in tandem to depict a workflow of data collection, data preprocessing, convolutional neural_network-based feature extraction, and machine learning-based analytics. This integrated system ensures that generating scorecards becomes smoother while interpreting umpire signals in real-time is improved to ensure fair, unbiased, and reliable decision-making in cricket. This approach provides a comprehensive solution, thanks to the technologies, which addresses the fast-changing cricket and technological integration [2].

This innovative procurement application is purposefully created to find a new way of using machine learning in order to revolutionize the formal or traditional experiences of cricket scorekeeping. By incorporating the development of remarkably sophisticated models upon vast information datasets, the automated system will ensure the accurate parsing of difficult game designs and umpire gestures, allowing the upgrading of scorecards and on-the-spot decisions. This not only eases the burden on statisticians and broadcasters but, more importantly, enhances the viewers experience with quick and correct information. Through the application of the open copy of CNNs, the modern sport's next development is this inherency of new technology into critical functionality with the anti-faulty reporting of data connected to the respective games. At the same time, attempting to automate scorecards has coincided with the ever-rising need and demand for more data-driven approaches to sporting analyses, which would further bring fair play and informed analytics. To accept this technology is to begin the process of refining cricket and placing it right into the cutting-edge of sporting change-taking modern audiences and stakeholders with it.

2. Literature Review

The current paper describes an automated online cricket scorecard that interprets umpire gestures using a vision and machine learning-based approach. The logistic regression model used for gesture classification was trained with 1,000 sample images of the 5 different gestures. Pre-processing was done on the images, converting them to monochrome and resizing them to 20x20 pixels for further usage in model training. The region of interest where computer vision recognizes gestures is defined using a Haar-cascade classifier. The accuracy and efficiency of this algorithm were checked against the training and testing data presented. The results show that this method is simple yet an efficient approach for scorecard automation in cricket, short-gaming the long-duration matches while doing away with the need for umpires to put on special gloves with a sensor integrated into them [1].

A hybrid ML model for cricket video summarization, designed for the peculiarity of cricket rules and long-duration matches, has been presented in the study. The model combines the Stacked Gated Recurrent Neural Network with Attention Module (SGRNN-AM) and a new hybrid rotation forest deep belief network (HRF-DBN). SGRNN-AM is used to analyze audio

in order to extract exciting clips on the basis of an adaptive threshold and speech-to-text framework, and detects key moments such as sixes, fours, wickets by focusing on significant features through an attention module. Scenes inside these clips are classified by the HRF-DBN. The dataset used for model evaluation consists of various collections of cricket videos, although the paper fails to mention any specifics about the dataset. The model achieved high-performance metrics leading to a precision of 96.82% and an accuracy of 96.32%, signifying its effectiveness in cricket video summarization by detecting key highs and events correctly [3].

To make a cricket ball/swing more predictable, the methodology in cricket must constantly be developed. It goes through the research papers, evaluating constraints in three pieces: tracking of player's performance, match simulation, and team selection. Different approaches and solutions are evaluated to write strengths and drawbacks down [4].

The paper deals with the development of a system to analyze cricket shots using 2D Convolutional Neural Networks and automatically classify for three main classes of shots. Such a system was able to classify various cricket shots with 91.5% accuracy overall. The model was validated against unseen data. The authors highlighted the utility of using 2D CNNs in sports analysis and there is a possibility for improvement if the dataset is diversified and augmented with representative data. The high accuracy of the model shows that deep learning can indeed prove very beneficial in the area of sports technology, with applications spanning into player movement tracking using sensor data [5].

The paper presents an application of the HTC-RCNN for cricket scorecard analysis. The HTC-RCNN integrates the architectures of Cascade and Mask R-CNN, which is applied for automatically extracting and evaluating detailed statistics from cricket scorecards, including player names, runs scored, wickets taken, and other information. It has exhibited an accuracy in identifying team name, player name, and runs with an average of 96.1%, 98.7%, and 97.3%, respectively. The results demonstrate that the HTC-RCNN model outperforms the competing methods regarding accuracy and efficiency, making it potentially very useful in automating cricket match analyses for highlights, player evaluations, and statistical insights [6].

The researchers present a CNN model that recognizes and classifies cricket events based on umpire action gestures. The newly introduced SNWOLF dataset consists of 1040 images of umpire action gestures corresponding to 6 cricket events: SIX, NO BALL, WIDE, OUT, LEG BYE, and FOUR. The CNN was trained with 80% of the SNWOLF dataset and tested on 20% of samples. It reached a good enough model accuracy with an overall standard accuracy of 98.20% over the test set. The output results were validated using the confusion matrix, which helps in deriving detailed metrics such as Precision, Recall, and F1 score for each class. High accuracy and low cross-entropy losses indicate the robustness of the model and its potential as a cricket sport highlight automatic generation tool [7].

Another study looks at detecting and recognizing gestures made by cricket umpires. The approach takes HOG feature extraction with NL-SVM classifier. The dataset used in that study consisted of 193,000 frames from the One Day International cricket match video. A smaller subset of 80 images of umpire non-action was also used to train the model. The accuracy level attained by the model stood at 97.95%. It signifies the model is pretty good at discerning between umpire gestures and non-gestures. The true positive classification accuracy of umpire action images was 98.60%. The manual frame selection for feature extraction in this method may lead to bias and restrict its relevance on other datasets [8].

This paper provides a framework to recognize cricket shots and detect cricket gestures by utilizing MediaPipe, LSTM networks, and CNN. A custom dataset was created for cricket shot detection, and the SNOW dataset was used for umpire and non-umpire classification and umpire gesture detection. The system yields 86.93% for umpire and non-umpire classification, and 80.51% for umpire gesture detection, in addition to 83.34% for cricket shot detection. The model successfully recognizes gestures including No Action, Wide, Six, No Ball, and Out, while shots detected include Pre Stance, Stance, Pull Shot, and Straight Drive. A limitation in this study would be that it will be very difficult to find large enough datasets to enhance performance. The study suggests future work could consider advanced CNN architectures, other features, and real-time tracking with the aim of improving the accuracy and robustness of the system [2].

Three popular deep learning models-VGG16, ResNet50V2, and MobileNetV2-are applied. These models are goal-honed by pre-training to detect numerous objects from the large dataset ImageNet. The models were then internet-based for posture recognition using SVM and Naïve Bayes classifiers. A novel dataset with 350 images was created specifically for this research, recognizing five different signals given by umpires in cricket: Six, No Ball, Out, Wide, and None. An accuracy of 80%, during 10-Fold Cross-validation and 70.83%, during Leave-One-Out validation was achieved by the MobileNetV2 model using the SVM classifier. Among the latter, ResNet50V2 performed best when combined with the SVM classifier, respectively obtaining 70.05%, 68.00%, and 67.69% as scores for average precision, recall, and F1. The study suggests that deep learning has provided a basic approach to simultaneously impose video content analysis in cricket, hence holding promise for future mobile applications for the same [9].

The article presents an innovative system that employs machine learning to detect umpire poses in cricket matches, enabling the generation of video highlights. The model is based on the Decision Tree classification technique, utilizing simple features extracted from randomly initialized linear models, namely Linear Regression and Ridge Regression models. The employed dataset known is as SNOW, containing images of umpire signaling events such as Six, No Ball, Out, and Wide.Noteworthy with high accuracy reported, the model that used SVM with features obtained from the first fully connected layer (fc1) of VGG19. Results evidence the system's efficiency in identifying images of umpires and their poses with good accuracy performance. However, why misclassifications happen is not yet completely understood. The article also points out the issues about misclassifications caused by features like hats or players' actions that could be mistaken as signals of umpire [10].

The article presents an innovative system that employs machine learning to detect umpire poses in cricket matches, enabling the generation of video highlights. The model is based on the Decision Tree classification technique, utilizing simple features extracted from randomly initialized linear models, namely Linear Regression and Ridge Regression models. The employed dataset known is as SNOW, containing images of umpire signaling events such as Six, No Ball, Out, and Wide.Noteworthy is the SVM model, with features enacted from the first fully connected layer (fc1) of the VGG19 network, showing the highest accuracy reported. The results will show the efficacy of the system in effectively identifying images of umpires and their poses with good performance accuracy. However, some misclassified dispositions are confounded. The authors also acknowledge limitations on the aspects of hat or player actions that could be interpreted over the basic umpires' signals [11].

An article describes a novel methodology to automate updating a cricket scoreboard, executed through the detection of umpire gestures with support vector machine (SVM) classifiers and a model based on the Inception V3 network.The network engages Inception V3, which had previous training on ImageNet, for the extraction of features. The system applies two SVM classifiers for image discrimination between umpires and non-umpires and identification of the umpire gestures like 'Six', 'Wide', 'No ball', and 'Out'. The dataset comprises images split 50/50 for training and testing, Although the exact figures are not provided, the system has achieved satisfactory preliminary results, as evidenced by the test accuracy calculated on unseen data. However, there are limitations to the system, including issues with misclassification, particularly in distinguishing between similar poses, and the current inability to recognize gestures other than the four that have been trained [12].

The proposed model worked for basketball official referee signals (ORSs), which consists of sixty-five different types of gestures. The model accomplished an accuracy of 96.6%, presenting a fluctuation of 2.0%. To evaluate its performance, the model was tested on the basketball official referee signals (ORSs) dataset, which consists of sixty-five different types of gestures The study also considered a partially supervised plan to improve performance with limited labelled information, but some limitations are not pointed at explicitly, such as the need for diverse datasets to generalize the applicability of the model or challenges of real-world deployment. Moreover, the investigation considers a partially supervised approach to enhance performance in the midst of limited labelled information [13].

A study introduces a gesture recognition technique that applies a 2-CNN architecture to extract features from Enhanced Depth Motion Maps (eDMM) and Static Pose Maps (SPM), which are then fused and classified with the aid of an Artificial Neural Network (ANN). For background removal, the Otsu method was applied. The methodology was tested on the Chalearn IsoGD dataset and the NATOPS dataset, with the proposed scheme outperforming the baseline method, with a test accuracy value of around 24.19%, achieving an accuracy value of 43.9% on the Chalearn IsoGD dataset. On the other hand, for the NATOPS dataset, the result was competitive with an accuracy of 83.47%. According to the results obtained, the combined use of eDMM and SPM enhanced gesture classification performance, in which the ANN classifier exhibited better performance and faster convergence compared to an SVM classifier [14].

In a broad manner, the literature survey of the paper titled Gesture Recognition would refer to exploring the methodology presented for human-hand gesture recognition using a model-based approach. The paper describes a finite state machine that models four qualitative distinct substrates of a gesture, a common approach in gesture recognition research to deal with the temporal aspect of gestures. This expands into the area, including fingertip tracking across multiple frames to compute motion trajectories, which is more or less in line with the available literature for tracking and motion analysis provided it's the key in understanding the dynamics of hand movements. Gesture recognition is a very important and decisive step in gestural recognition because it makes the definition of gesture-associated boundaries clear for subsequent analyses. Representing gestures through a list of vectors and matching their respective vectors to available models of discrete gestures, based on table lookup by vector displacements, resonates well with the traditional approaches to pattern recognition and machine learning techniques found in the available literature. The clarifying contributions showed several results; the recognition of seven gestures is very clear, sampled images at 4Hz on a SPARC-1 without any special hardware make the approach feasible in a low-resourced context. This aligns with the ongoing research in making gesture recognition more accessible and efficient. The literature survey, therefore, would place this paper within the broader context of gesture recognition research, noting its specific contributions and methodologies in relation to existing work in the field.

In the domain of motion identification, the utilization of sensing technology has been a fundamental element for progress, with the detection of magnetic fields serving as a notable technique for both indoor positioning and as an additional approach for devices that recognize motion. The investigation conducted by the respective authors introduces an innovative system that takes advantage of multiple sensors to quantify the magnetic field of Earth, which is subsequently employed to gather data for the identification of motion through a one-dimensional convolutional neural network (1D CNN). Through this groundbreaking method, significant improvement in motion recognition has been shown, evidenced by experiments that achieved a recognition rate of almost 97% for the identification of standard letters in American Sign Language. The reliance of the system on Earth's magnetic field for detection simplifies the intricacy of the device in comparison to methods based on artificial magnetic fields, all the while not impeding the movements of the user's hand. The potential applications of this technology are enormous, with significant implications for the interaction between computers and humans in different domains such as virtual or augmented reality, healthcare sector, smart homes, and tools that assists individuals with various disabilities. Motion identification technology, as a component of human-computer interaction (HCI), is characterized by its convenience, and substantial interactivity, which make it an exceptional choice for natural and genuine interactions between humans and computers. The investigation underscores the future of motion recognition technology that utilizes magnetic sensing as a precise and dependable method that has the potential to revolutionize the manner in which we interact with digital environments and devices [15].

3. Proposed Method

3.1. Proposed Working Flow

Figure 1 presented the flowchart which illustrates a proposed methodology for the classification of cricket gameplay elements using machine learning techniques. The primary goal of this technique is to construct a resilient system for the identification and assessment of cricket gaming aspects by exploiting Convolutional Neural Networks (CNN). The process commences with the gathering of primary and secondary datasets. The main dataset is anticipated to be an extensive collection of raw datasets obtained from Kaggle [17], while the secondary dataset will be used as an additional source to enhance the variety and amount of data accessible for training our model. We merge and consolidate all the datasets into one single unified dataset on which the data preprocessing tasks are adequately conducted to ensure the cleansing, transformation,
and processing of data for training.

As has been stated, the kernel functionality relies on the design of the CNN model, which would have been shown to be occasionally hard to beat in regard to the test of image classification. We are confident that this model can reasonably discover and extract the hierarchical features from the data set after pre-processing. We can optimize the architecture for the CNN that reflects efficiency in learning with the least use of computation. After the training stage, our CNN will classify each data instance by some elocution such as Six, Four, Dead Ball, and many more based upon patterns and features it has learned while training. The classified items will be used in scoring, whose outcome depends on the different identified actions
under the gameplay.

The proposed method, therefore, aims, at length, at the development of a robust system for classification and scoring of the actions that are carried on during a cricket game by the use of Convolutional Neural Networks. Starting with the main data and the additional datasets, they will be combined into one big dataset and text-based preprocessed extensively. The kernel functionality in the methods itself is the effective use of the CNN model, which allows learning and extracting hierarchical features from the preprocessed data. An altered architecture of the CNN will be optimized for efficient learning, on one hand, and discrimination power on the other hand. After training, our CNN will write each data instance in such a way as to configure specific categories, like 'Six', 'Four', 'Revoke', and some class path, depending on the model and achieved feature extraction. The classified elements will serve to score, which gives a total score depending on the identified actions in the game around. Proposed method has potential to revolutionize classification and scoring in cricket gameplay elements, an action that would present to cricket gameplay analysts a highly accurate and efficient system for analysis.

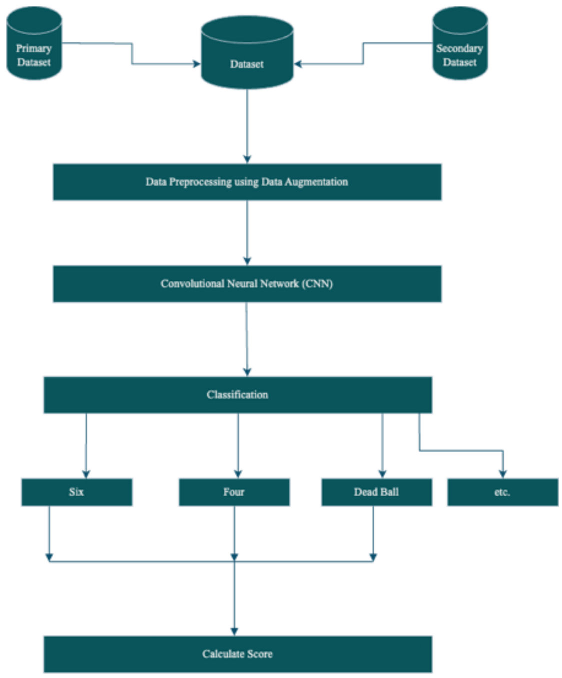


Figure 1: Flow Chart

The process of data analysis is started by collection of datasets. The dataset downloaded from kagle is converted into a unified dataset so that the diversity is ensured; and also, to improve the performance of the model. Data preprocessing is done to ensure images are in a consistent shape, normalized pixel values, and applying augmentation such as flipping and rotation to improve variability. This helps limits overfitting and can enhance model generalization. The dataset was split into train (70%), validation (15%) and test (15) sets. The CNN models extract information at a hierarchical level, with the first layers identifying edges/barriers and textures, while layers deepen can be trained to indicate more complex patterns (i.e. umpire gestures made by an official). The model training is conducted by the Adam optimizer and loss of sparse categorical cross-entropy. Accuracy trends are recorded in the training stage, and predictions on the final model can be checked against the actual labels to determine performance of the model. The trained model was able to achieve an acceptable level of precision for classifying gameplay. Soccer would be considered an higher-level precision model as there is the ability to detect variations differences in an umpire's gestures.

3.2. Proposed Sequential Algorithm for Model Train

1. **Data Initialization**
   - Define image size and batch size.
   - Load images from directory.
   - Extract class names from the dataset.
   - Split the dataset into training, validation, and testing sets depend on specified proportions.
   - Define a function **get_data** to perform the splitting with optional shuffling.

2. **Data Augmentation and Pre-processing**
   - Define a **resize_and_rescale** model to resize images and normalize pixel values.
   - Define a **data_augmentation** model for random flipping and rotation.
   - Define the input shape depend on batch size, size of image, and number of channels.
   - Define a **resize_and_rescale** model to resize images and normalize pixel values.
   - Define the number of classes based on the dataset.
3. **Model Definition**
   - Create a sequential model.
   - Add **resize_and_rescale** and **data_augmentation** models.
   - Incorporate convolutional layers along with ReLU activation and max pooling layers.
   - Add a flatten layer in order to modify the output to a 1D vector.
   - Add dense layers with ReLU activation for hidden units and softmax activation for the output layer (number of classes).

3 **Model Training and Evaluation**
   - Build the model with the defined input shape.
   - Print model summary.
   - Compile the model with Adam optimizer, sparse categorical cross-entropy loss, and accuracy metric.
   - Train the model with the training dataset for a specific number of epochs, while considering the batch size and incorporating validation data.
   - Evaluate the model on the testing dataset and print the accuracy scores.

4 **Prediction and visualization**
   - Take one batch of images and labels from the testing dataset.
   - Get the first image and its label.
   - Print the actual label based on class names.
   - Predict the label for the image using the model.
   - Print the predicted label.
   - Plot the training and validation accuracy curves from the training history.

3.3. Design and Implementation

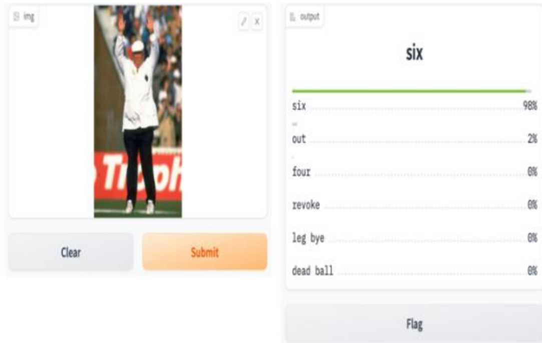
3.3.1 Dataset

The study utilizes a selection of photographs of cricket match umpires, which provides the basis for automating modifications to cricket scorecards through the identification of hand movements. The size of dataset is approximately 1600 images of different hand gestures.

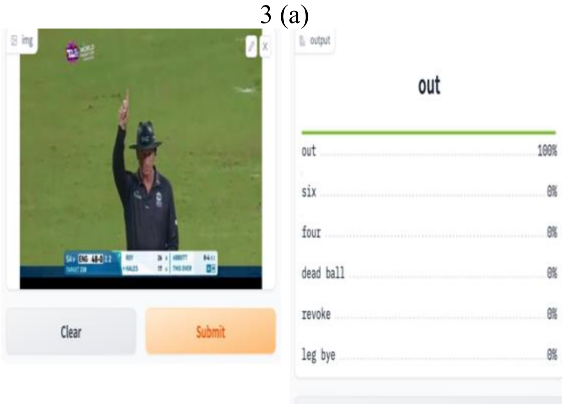
Figure 2: Dataset Sample

In figure 2, first sample image is displaying the 'no ball' gesture, second sample image is displaying the gesture of 'out', and third sample image is showing the 'Six' as gesture.

The testing of the application is done using input image of umpire decision is given to the system and system predict the output as final decision as mentioned in figure 3 (a - f).

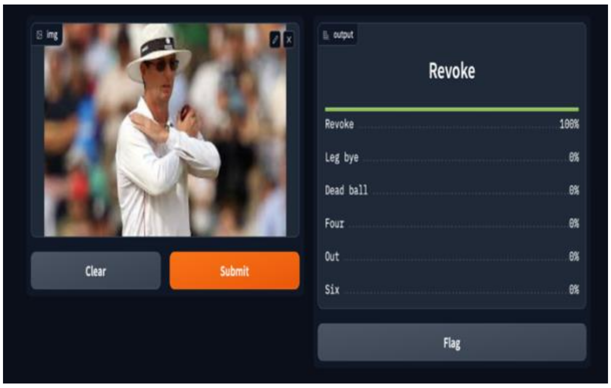


3 (a)

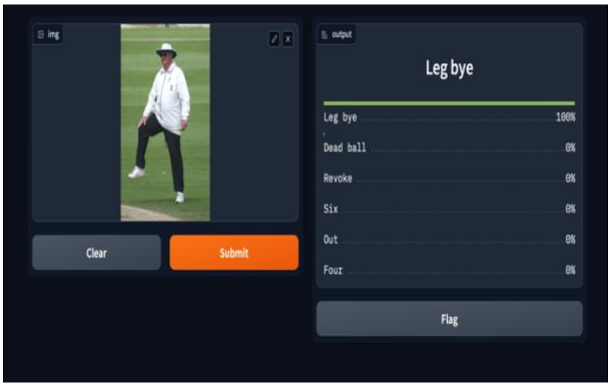


3 (b)

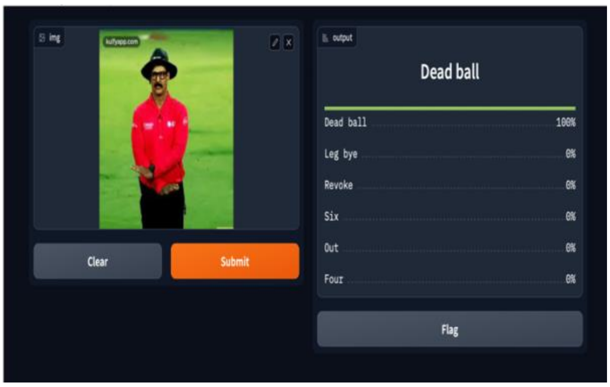
3 (c)



3 (d)



3 (e)



3 (f)

Figure 3 (a, b, c, d, e, f): Result Test Dataset Predicted with 98% Accuracy

In figure 3 (a, b) image depicts the intersection of human umpire gestures and machine learning in officiating. The umpire signals "Six" (in figure 3 (a)), and "out" (in figure 3 (b)) traditionally, while an CNN based model analyzes the scene using computer vision technology and predicts "six" with 98% precision (in figure 3 (a)) and in figure 3 (b), predicts "out" correctly by comparing it to a vast database of labeled dataset.

While in figure 3 (c to f) image depicts the intersection of human umpire gestures and machine learning in officiating. The umpire signals "four" (in figure 3 (c)), "Revoke" (in figure 3 (d)), "leg bye" (in figure 3 (e)), and "dead ball" (in figure 3 (f)).

4. Results and Discussions

4.1. Result of Proposed Model

The figures 4 represent the performance metrics of a proposed model over a validation process. Figure 4 likely shows two curves, one for training accuracy and another for validation accuracy (val_accuracy) across epochs during model training. These curves would illustrate how well the model predicts the correct labels for both the training data (Training accuracy) and unseen data (val_accuracy), with the goal being to maximize both values, indicating high generalization of the model.

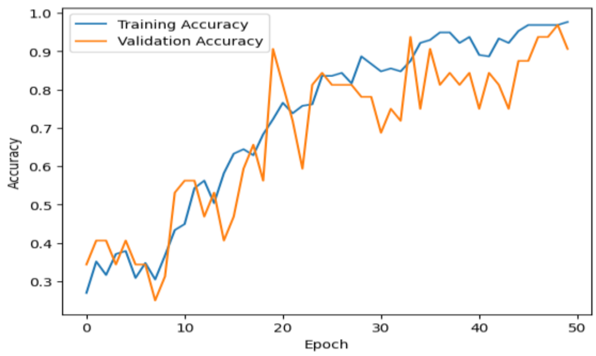


Figure 4: Training and Validation Accuracy of the Proposed Model

4.2. Comparison of Proposed Model with Existing

The table 2 and figure 5 provides a summary of performance of four predictive models: Logistic Regression, LSTM, Inception V3, and Proposd CNN is compared using key metrics: accuracy, precision, recall, and F1-score. Logistic Regression shows strong overall performance with 91.06% accuracy and closely aligned precision and recall around 88%, indicating a balanced ability in making correct predictions.

Table 2: Comparison of Proposed Model with Existing Work

Model	Accuracy	Precision	Recall	F1-Score
logistic Regression [1]	91.06%	88.10%	88.13%	88.13%
Long Short-Term Memory [2]	90%	89%	89%	89%
Inception V3 [16]	66.10%	75%	75%	75%
Proposed CNN	98.25%	97.23%	95.60%	97.10%

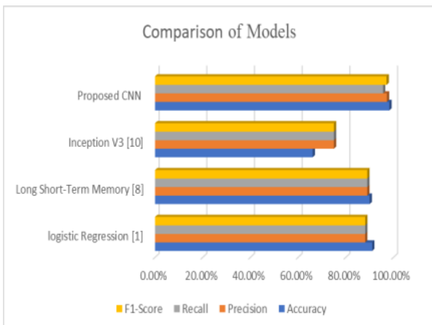


Figure 5: Comparison of Models

LSTM is slightly behind in accuracy at 90% but maintains consistent performance across all metrics at 89%, demonstrating its effectiveness in balancing true positive and negative predictions. Inception V3 lags with the lowest accuracy at 66.10% and uniform metrics at 75%, suggesting it struggles more in classifying cases accurately compared to the others. CNN shines as the top performer with the highest accuracy at 98.25%, and superior precision and F1-score above 97%, while its recall at 95.60% shows it slightly lags in identifying all relevant cases. This overview highlights the strengths and limitations of each model, emphasizing CNN's exceptional ability to predict accurately with minimal errors, and suggesting the careful selection of models.

4.3. Accuracy Testing

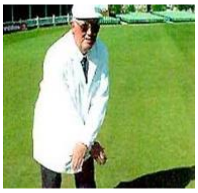
The table 3 represented the accuracy of empire decision images for various umpire signals, the similar type umpire signals are taken for averaging the accuracy of the particular signal.

Table 3: Manual Accuracy Testing

Umpire Signal Image	**Expected Output**	**Proposed Model Output**	**Prediction Status**
	Six	Six	Correct
	Six	Six	Correct
	Six	Six	Correct
	Six	Six	Correct
	Six	Six	Correct

Umpire Signal Image	**Expected Output**	**Proposed Model Output**	**Prediction Status**
	Six	Out	Incorrect
	Out	Out	Correct
	Out	Out	Correct
	Out	Out	Correct
	Out	Out	Correct
	Out	Out	Correct
	Four	Four	Correct
	Four	Four	Correct

Umpire Signal Image	Expecte d Output	Propose d Model Output	Predictio n Status
	Four	Four	Correct
	Four	Four	Correct
	Four	Four	Correct
	Revoke	Revoke	Correct
	Revoke	Revoke	Correct
	Revoke	Revoke	Correct
	Revoke	Revoke	Correct
	Revoke	Revoke	Correct
	Dead Ball	Dead Ball	Correct

Umpire Signal Image	Expecte d Output	Propose d Model Output	Predictio n Status
	Dead Ball	Dead Ball	Correct
	Dead Ball	Dead Ball	Correct
	Dead Ball	Dead Ball	Correct
	Dead Ball	Dead Ball	Correct
	Leg bye	Leg bye	Correct
	Leg bye	Leg bye	Correct
	Leg bye	Leg bye	Correct
	Leg bye	Leg bye	Correct

Umpire Signal Image	Expected Output	Proposed Model Output	Prediction Status
	Leg bye	Leg bye	Correct

Calculating average accuracy of umpire signals:

For each gesture Gi, where i ranges from 1 to the total number of unique gestures:

Accuracy of umpire single is

$$\text{Accuracy}_{G_i} = \frac{\sum_{j=1}^{n_i} \text{Correct}_j}{n_i}$$

- ni is the total number of predictions for gesture Gi.
- Correctj is 1 if the jth prediction for gesture Gi is correct, and 0 otherwise.

Calculate the Overall Average Accuracy:
To find the overall average accuracy, sum up the accuracies for each gesture and then divide by the total number of unique gestures:

$$Overall\ Average\ Accuracy = \frac{\sum_{i=1}^{N} Accuracy\ G_i}{N}$$

Where:
N is the total number of unique gestures.
Calculations as per Table 3: Manual Accuracy Testing:

$$Overall\ Average\ Accuracy = \frac{0.833 + 1 + 1 + 1 + 1 + 1}{6} = \frac{5.833}{6} \approx 0.9722$$

As per manual testing of proposed CNN model, the overall average accuracy across all gestures is approximately 97.22%.

4.4. Implications

The proposed model improves decision-making in cricket by automating umpiring calls with a reported 98% accuracy rate which in return decreases human officiating errors. But still, it helps to reinforce match decision reliability. Automated identification of match components makes real-time scoring of information easier for scorers and creates more accurate and reliable digital scorecards. This is beneficial in lower entry-level cricket matches as decision accountability was not as strict in the past creating errors to build. The approach is scalable and has potentially unlimited applicability including being used for other sports requiring gesture intervention for officiating during the contest (i.e., baseball, tennis). Additionally, it could be fine-tuned for T20 matches, ODIs, and Test matches, each with its own associated scoring methods. The modeling comparisons to Logistic Regression, LSTM, and Inception V3 demonstrated that the CNN model performed superiorly in terms of accuracy, predictive validity, and overall performance. Future work could focus on video analysis, and potentially hybrid models or incorporating machine learning with respect to accuracy with a multitude of variables.

5. Conclusion and Future Scope

This study has successfully demonstrated the potential of utilizing the Convolutional Neural Networks (CNN) model for the analysis & classifications of various cricket gameplay elements, gaining an average accuracy of 98%, which is notably higher than other methodologies examined. showcasing a significant improvement over existing methods with its ability to accurately categorize game actions and compute scores. The research focuses the model's effectiveness in leveraging the machine learning for analytics in sports, with comparative analysis highlighting its superiority in terms of accuracy, precision, against the available traditional approaches. Looking forward, the exploration of advanced neural network architectures, expansion of datasets to encompass a wider variety of gameplay scenarios, and the integration of real-time analysis capabilities present promising avenues for future work. Additionally, the potential for interdisciplinary applications and the imperative for ethical considerations in the automation of sports analytics outline the broader implications and opportunities for further research in this field, suggesting an exciting frontier for the intersection of Artificial Intelligence and sports.

Declarations

Funding: None

Conflicts of interest: The authors confirm that they have no competing interests or conflicts of interest.

Availability of data and material: As no datasets were generated or analyzed during the current study, data sharing is not applicable to this article.

Code availability: The code for implementation is available upon request, subject to privacy and other restrictions.

References

[1] Md. A. Shahjalal, Z. Ahmad, R. Rayan, and L. Alam, "An approach to automate the scorecard in cricket with computer vision and machine learning," in *2017 3rd International Conference on Electrical Information and Communication Technology (EICT)*, Dec. 2017, pp. 1–6. doi: 10.1109/EICT.2017.8275204.

[2] V. Kumar, M. Pruthi, A. Katyal, R. Semwal, and A. professor, "Cricket Gesture Detection Using LSTM, Media Pipe And CNN," *Journal of Xidian University*, vol. 7, no. 09, Oct. 2023, doi: 10.37896/jxu17.9/068.

[3] H. Shingrakhia and H. Patel, "SGRNN-AM and HRF-DBN: a hybrid machine learning model for cricket video summarization," *Vis Comput*, vol. 38, no.

7, pp. 2285–2301, 2022, doi: 10.1007/s00371-021-02111-8.

[4] M. M. Hatharasinghe and G. Poravi, "Data Mining and Machine Learning in Cricket Match Outcome Prediction: Missing Links," in *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, 2019, pp. 1–4. doi: 10.1109/I2CT45611.2019.9033698.

[5] R. Huang, H. Wang, F. Liu, and W. W. Guo, "Design of cricket automatic control system based on computer vision technology," in *Proceedings - 2019 International Conference on Robots and Intelligent System, ICRIS 2019*, Institute of Electrical and Electronics Engineers Inc., Jun. 2019, pp. 26–29. doi: 10.1109/ICRIS.2019.00016.

[6] S. Ray and U. Singh, "Hybrid Task Cascade Region-Based Convolutional Neural Network is used to Analyse Cricket Scores."

[7] S. Nandyal and S. L. Kattimani, "Umpire gesture detection and recognition using HOG and non-linear support vector machine (NL-SVM) classification of deep features in cricket videos," in *Journal of Physics: Conference Series*, IOP Publishing Ltd, Nov. 2021. doi: 10.1088/1742-6596/2070/1/012148.

[8] S. Nandyal, S. L. Kattimani, and P. G. Halakatti, "Cricket Event Recognition and Classification from Umpire Action Gestures using Convolutional Neural Network." [Online]. Available: www.ijacsa.thesai.org

[9] W. Samaraweera, S. Premaratne, and A. Dharmaratne, "Deep Learning for Classification of Cricket Umpire Postures," in *Communications in Computer and Information Science*, Springer Science and Business Media Deutschland GmbH, 2020, pp. 563–570. doi: 10.1007/978-3-030-63823-8_64.

[10] A. Ravi, H. Venugopal, S. Paul, and H. R. Tizhoosh, "A Dataset and Preliminary Results for Umpire Pose Detection Using SVM Classification of Deep Features," *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, pp. 1396–1402, Jul. 2018, doi: 10.1109/SSCI.2018.8628877.

[11] M. Tiwari, V. Chaurasia, E. A. Siddiqui, V. Patel, A. Kumar, and M. Patankar, "Enhanced Image Compression Using Fractals and Principle Component Analysis," in *1st IEEE International Conference on Innovations in High Speed Communication and Signal Processing, IHCSP 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 502–507. doi: 10.1109/IHCSP56702.2023.10127168.

[12] D. T. John, K. S. Kumar, V. T. Nair, P. Visakh, and B. R. Poorna, "Cricket Scoreboard Automation using Umpire Gestures," *International Journal of Research in Engineering, Science & Management*, vol. 2, no. 7, pp. 344–348, Jul. 2019, Accessed: Feb. 24, 2024. [Online]. Available: https://www.ijresm.com/Vol.2_2019/Vol2_Iss7_July19/IJRESM_V2_I7_80.pdf

[13] T. Y. Pan, C. Y. Chang, W. L. Tsai, and M. C. Hu, "Multisensor-Based 3D Gesture Recognition for a Decision-Making Training System," *IEEE Sens J*, vol. 21, no. 1, pp. 706–716, Jan. 2021, doi: 10.1109/JSEN.2020.3012887.

[14] Z. Zhang, S. Wei, Y. Song, and Y. Zhang, "Gesture Recognition Using Enhanced Depth Motion Map and Static Pose Map," in *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, IEEE, May 2017, pp. 238–244. doi: 10.1109/FG.2017.38.

[15] B. Shi, X. Chen, Z. He, H. Sun, and R. Han, "Research on Gesture Recognition System Using Multiple Sensors Based on Earth's Magnetic Field and 1D Convolution Neural Network," *Applied Sciences (Switzerland)*, vol. 13, no. 9, May 2023, doi: 10.3390/app13095544.

[16] A. Ravi, H. Venugopal, S. Paul, and H. R. Tizhoosh, "A Dataset and Preliminary Results for Umpire Pose Detection Using SVM Classification of Deep Features," 2018. [Online]. Available: https://drive.google.com/drive/folders/1ljDIz69mJqDzBlUxABP0c34NgdynzWPX

[17] "Cricket umpire keypoints." [Online]. Available: https://www.kaggle.com/datasets/harold511/cricket-umpire-keypoints

Chapter B

Plagiarism Report

Next-Gen Cricket Scorecard Automation Using Deep Neural Networks

ORIGINALITY REPORT

10%	7%	6%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	arxiv.org Internet Source	1%
2	R. N. V. Jagan Mohan, B. H. V. S. Rama Krishnam Raju, V. Chandra Sekhar, T. V. K. P. Prasad. "Algorithms in Advanced Artificial Intelligence - Proceedings of International Conference on Algorithms in Advanced Artificial Intelligence (ICAAAI-2024)", CRC Press, 2025 Publication	1%
3	www.flll.uni-linz.ac.at Internet Source	<1%
4	Poonam Nandal, Mamta Dahiya, Meeta Singh, Arvind Dagur, Brijesh Kumar. "Progressive Computational Intelligence, Information Technology and Networking", CRC Press, 2025 Publication	<1%
5	www.mdpi.com Internet Source	<1%
6	"Computer Vision and Image Processing", Springer Science and Business Media LLC, 2024 Publication	<1%
7	aiforsocialgood.ca Internet Source	<1%

13/06/25

*% detected as AI

AI detection includes the possibility of false positives. Although some text in this submission is likely AI generated, scores below the 20% threshold are not surfaced because they have a higher likelihood of false positives.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.