DEEP LEARNING-BASED SPORTS SHOT CLASSIFICATION SYSTEM

A PROJECT REPORT

Submitted by

DHARSHIKA S (211421104057)

HARI PRIYA R (211421104086)

LOGESHWARI S (211421104146)

in partial fulfillment for the award of the degree of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2025

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report **“DEEP LEARNING-BASED SPORTS SHOT CLASSIFICATION SYSTEM”** is the bonafide work of **“DHARSHIKA S (211421104057), HARI PRIYA R (211421104086), LOGESHWARI S (211421104146)”** who carried out the project under my supervision.

SIGNATURE	**SIGNATURE**
Dr L.JABASHEELA M.E., Ph.D.,	**Dr M S.VINMATHI M.E., Ph.D.,**
PROFESSOR AND HEAD,	**SUPERVISOR**
	PROFESSOR
Department of Computer Science and Engineering, Panimalar Engineering College, Chennai - 123	Department of Computer Science and Engineering, Panimalar Engineering College, Chennai - 123

Certified that the above candidate(s) was examined in the End Semester Project Viva-Voce Examination held on ....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

DECLARATION BY THE STUDENT

We DHARSHIKA S (211421104057), HARI PRIYA R (211421104086), LOGESHWARI S (211421104146) hereby declare that this project report titled “DEEP LEARNING-BASED SPORTS SHOT CLASSIFICATION SYSTEM”, under the guidance of DR M S VINMATHI is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

DHARSHIKA S

HARI PRIYA R

LOGESHWARI S

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D**., for his fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express our heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Dr. KAVITHA SUBRAMANI, M.E., Ph.D.,** and **Dr. M S. VINMATHI M.E., Ph.D.,** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

DHARSHIKA S

HARI PRIYA R

LOGESHWARI S

31.03.2025

To Whomsoever It May Concern

This is to certify that **DHARSHIKA S (211421104057), HARI PRIYA R (211421104086), LOGESHWARI S (211421104146)**, a student of final year **B.E. COMPUTER SCIENCE AND ENGINEERING** of **"PANIMALAR ENGINEERING COLLEGE"** has completed their major project with great success at our concern, under the Title: **"DEEP LEARNING-BASED SPORTS SHOT CLASSIFICATION SYSTEM**" from **DECEMBER 2024** to **MARCH 2025.**

Their project is found to be relevant regarding their stream, and they have submitted a copy of the project report to us. During their Project period, we found them sincere & hard-working & possessing good behaviour and a moral character.

We wish them grand success in future endeavours.

For SPIRO PRIME TECH SERVICES,

M.SAMPATH KUMAR

MANAGER

ABSTRACT

Sports analytics have become an essential component in improving athletic performance and enhancing coaching strategies. This project presents a cutting-edge Sports Shot Classification System that utilizes deep learning and Convolutional Neural Networks (CNNs) for accurate shot classification. The system employs TensorFlow to build and deploy models, ensuring robust scalability and efficiency throughout both training and operational phases. By optimizing CNN architectures for shot classification and employing advanced strategies like data augmentation, hyperparameter tuning, and deep learning techniques, the system delivers high precision in recognizing various sports shots. Its real-time capabilities enable instant analysis of gameplay, offering valuable insights for coaches, analysts, and broadcasters. This system not only improves the accuracy of shot detection but also enhances decision-making processes by providing a detailed breakdown of player performance. By automating the shot classification process, Sports Shot Classification System reduces the reliance on manual observation, ensuring faster and more reliable analysis.With the ability to scale efficiently, the system can be deployed in both training environments and live event settings, offering significant improvements to sports performance evaluations and broadcasting experiences.This framework holds the potential to revolutionize sports analysis, coaching, and broadcasting, offering automated and intelligent classification across different sports contexts.

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- AI - Artificial Intelligence
- CNN - Convolutional Neural Network
- Vit-Vision Transformer
- RNN-Recurrent Neural Network
- VGG-Visual Geometry Group
- GRU- Gated Recurrent Unit
- OpenCV-Open Source Computer Vision Library
- FPS-Frames Per Second
- ReLU-Rectified Linear Units
- GPU-Graphics Processing Unit
- LAN-Local Area Network
- HDD-Hard Disk Drive
- SSD-Solid State Drive

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

CHAPTER 1

INTRODUCTION

Advances in data collecting and analysis technology have led to a considerable evolution in sports analytics in the past few years. Accurate classification of sports shots is crucial for enhancing sports analytics, coaching methodologies, and broadcasting, as this offers vital information about gameplay dynamics and athlete performance, essential for making informed decisions. Traditional techniques for shot classification often rely on manual labeling or basic rule-based systems. While useful to some extent, these methods have significant limitations, including being labor-intensive, time-consuming, and inflexible in handling the complexities and variations inherent in different sports. Recent advancements in Convolutional Neural Networks(CNNs), a kind of deep learning, have demonstrated significant potential in tackling these issues. CNNs immediately learn and extract relevant features from data, achieving high accuracy in classification tasks and adapting to the complexities of various sports. This study presents an innovative deep learning framework for the classification of sports shots using CNNs developed with TensorFlow.

Our approach includes exploring a range of optimized CNN architectures, employing data augmentation techniques, incorporating advanced deep learning methods such as dropout, batch normalization, transfer learning, and meticulously tuning hyperparameters to enhance model efficacy. By leveraging TensorFlow, we ensure robust scalability and efficiency during the model's training and deployment stages. Our proposed framework represents a noteworthy development in sports analytics, providing an opportunity to transform sports analysis, broadcasting, and coaching practices by automating and sophisticating shot classification. This, in turn, seeks to offer deeper information about athlete performance as well gameplay strategies, ultimately contributing to the evolution of sports analytics.

1.1. OVERVIEW OF THE PROJECT

The study focuses on developing an innovative deep learning-based framework for sports shot classification using Convolutional Neural Networks (CNNs) and TensorFlow. Accurate classification of sports shots is highlighted as essential for enhancing sports analytics, coaching methodologies, and broadcasting. Traditional manual and rule-based systems are labor-intensive and inflexible, which has driven the adoption of CNNs due to their ability to automatically extract features and adapt to various sports complexities. This framework leverages optimized CNN architectures, including AlexNet, VGG16, and LeNet, alongside techniques like data augmentation, transfer learning, and hyperparameter tuning. The implementation ensures robust scalability and efficiency during training and deployment. Extensive datasets of sports shot images are preprocessed, augmented, and categorized into distinct shot types to improve classification accuracy. Among the models tested, AlexNet showed the highest performance with its superior feature extraction and training stability. The project aims to automate sports shot classification, providing deeper insights into athlete performance and gameplay strategies, thereby advancing sports analysis, coaching, and broadcasting technologies.

1.2. PROBLEM STATEMENT

Accurate classification of sports shots is vital for improving sports analytics, coaching methodologies, and broadcasting by providing critical insights into gameplay and athlete performance. Traditional classification techniques, such as manual labeling or rule-based systems, are labor-intensive, time-consuming, and lack the adaptability to handle the complexities of various sports. Existing methods struggle with diverse variables like shot angles, player movements, and environmental factors, making them insufficient for practical applications in sports.
The limitations of manual processes underscore the need for an automated and

efficient classification system capable of handling large datasets with high accuracy. Convolutional Neural Networks (CNNs), a deep learning approach, have demonstrated remarkable potential by automatically learning complex features and delivering superior classification results. Developing a robust framework leveraging CNNs and advanced techniques like data augmentation, transfer learning, and TensorFlow can address the shortcomings of traditional methods, transforming sports analytics. Automated shot classification can offer deeper insights into athlete performance and gameplay strategies, enabling data-driven decision-making and strategic improvements. Leveraging modern frameworks like TensorFlow ensures that the proposed solution is scalable and efficient, making it practical for widespread application across diverse sports.

1.3. NEED FOR THE PROJECT

In the domain of sports analytics, accurate classification of sports shots is vital for improving analytics, coaching methodologies, and broadcasting by providing critical insights into gameplay dynamics and athlete performance. However, traditional methods, such as manual labeling and rule-based systems, are labor-intensive, time-consuming, and lack adaptability to handle the complexities of various sports, including diverse angles, player movements, and environmental conditions. Deep learning-based models, particularly Convolutional Neural Networks (CNNs) implemented with frameworks like TensorFlow, offer a transformative solution.

These models provide high accuracy and scalability while automating the process of shot classification. By leveraging techniques like data augmentation, hyperparameter tuning, and transfer learning, such a framework can effectively address the challenges posed by traditional methods, paving the way for enhanced sports analytics, coaching, and broadcasting practices.

1.4 .OBJECTIVE OF THE PROJECT

This project aims to develop an advanced deep learning framework for accurate sports shot classification, enhancing analytics, coaching, and broadcasting. By leveraging optimized CNN architectures like AlexNet, VGG16, and LeNet, it ensures precise shot identification while addressing challenges such as varying angles, player movements, and environmental factors. Performance is enhanced through data augmentation, transfer learning, dropout, and hyperparameter tuning. Automating sports analytics replaces manual efforts, streamlining decision-making, while TensorFlow enables scalable deployment for real-world applications. Ultimately, the system provides valuable gameplay insights, helping coaches and analysts refine strategies and optimize athlete performance.

1.5. SCOPE OF THE PROJECT

The Sports Shot Classification System automates sports shot classification, enhancing analytics, coaching, and broadcasting.

1. **Advanced Classification:** Utilize CNNs (AlexNet, VGG16, LeNet) to classify shots based on body posture, execution, and ball trajectory.
2. **Real-Time Analysis:** Enable instant classification and feedback for live performance assessment.
3. **Multi-Sport Adaptability:** Handle dynamic movements, angles, and conditions across various sports, integrating with analytics platforms and training centers.
4. **Optimized Performance:** Apply data augmentation, transfer learning, and hyperparameter tuning with TensorFlow for accuracy and scalability.

CHAPTER 2
LITERATURE SURVEY

CHAPTER 2

LITERATURE SURVEY

2.1. TITLE: Visual Analytics of Off-ball Movements in Basketball

NAME OF THE PUBLISHER: Yihong Wu.

YEAR OF PUBLICATION: 2023

OBJECTIVE: The paper aims to analyze and visualize off-ball player movements in basketball using advanced tracking data, providing insights into player positioning, spacing, and tactical decision-making.

INFERENCE: The study reveals that off-ball movements significantly impact team performance, influencing defensive strategies, offensive efficiency, and overall game dynamics through spatial awareness and movement patterns.

2.2. TITLE: Cricket Batting Shots Classification with Vision Transformer Network

NAME OF THE PUBLISHER: A. Dey

YEAR OF PUBLICATION: 2024

OBJECTIVE: This paper explores the use of Vision Transformer (ViT) for classifying cricket batting shots, leveraging self-attention mechanisms for feature extraction and improved accuracy.

INFERENCE: The study demonstrates that ViT outperforms traditional CNNs in recognizing cricket shots, capturing finer spatial dependencies for better shot classification.

2.3. TITLE: Classification of Cricket Shot from cricket Videos using Deep Learning Models

NAME OF THE PUBLISHER: Venkatesh B R

YEAR OF PUBLICATION: 2024

OBJECTIVE: The paper investigates deep learning models, such as CNNs and RNNs, for classifying cricket shots from video sequences, focusing on temporal and spatial feature extraction.

INFERENCE: The study shows that deep learning models effectively classify cricket shots by capturing motion dynamics, improving automated sports analytics and coaching insights.

2.4. TITLE: Similar Sports Play Retrieval with Deep Reinforcement Learning

NAME OF THE PUBLISHER: Zheng Wang

YEAR OF PUBLICATION: 2023

OBJECTIVE: This research proposes a deep reinforcement learning-based approach to retrieve and compare similar sports plays by analyzing player movements and game patterns.

INFERENCE: The findings indicate that reinforcement learning enhances play retrieval accuracy, aiding in tactical analysis, coaching strategies, and performance evaluation.

2.5. TITLE: Optimized Deep Learning-Based Cricket Activity Focused Network

NAME OF THE PUBLISHER: Ahmad et al.

YEAR OF PUBLICATION: 2023

OBJECTIVE: The study develops an optimized deep learning network to detect and analyze cricket activities, improving efficiency in recognizing player actions and game events.

INFERENCE: The proposed model enhances cricket activity recognition by optimizing feature extraction and computational efficiency, benefiting performance analysis and sports broadcasting.

2.6 TITLE: Shot-Net: A CNN for Classifying Cricket Shots

NAME OF THE PUBLISHER: Foysal et al.

YEAR OF PUBLICATION: 2019

OBJECTIVE: This paper presents Shot-Net, a specialized CNN model designed to classify cricket shots based on spatial features from images and videos. **INFERENCE:** The study shows that Shot-Net effectively distinguishes between cricket shots, achieving high accuracy and improving automated sports analysis.

2.7. TITLE: CricShotClassify: Classifying Batting Shots Using CNNs and GRUs

NAME OF THE PUBLISHER: Sen et al.

YEAR OF PUBLICATION: 2021

OBJECTIVE: The research introduces a hybrid CNN-GRU model for classifying batting shots, leveraging CNNs for feature extraction and GRUs for sequential dependencies.

INFERENCE: The findings demonstrate that combining CNNs with GRUs improves shot classification, particularly in capturing temporal variations in cricket strokes.

2.8. TITLE: Learning Cricket Strokes Using Visual Word Sequences

NAME OF THE PUBLISHER: Gupta and Muthiah

YEAR OF PUBLICATION: 2023

OBJECTIVE: This study applies visual word sequence-based learning to recognize cricket strokes, treating shot classification as a sequence prediction problem.

INFERENCE: The approach enhances shot recognition by learning motion patterns as visual words, providing a new perspective on cricket analytics.

2.9. TITLE: Transformer-Based Cricket Shot Classification Model

NAME OF THE PUBLISHER: Azhar et al.

YEAR OF PUBLICATION: 2023

OBJECTIVE: This research explores Transformer architectures for cricket shot classification, leveraging self-attention to capture complex spatial and temporal relationships.

INFERENCE: The study concludes that Transformers outperform CNNs in shot classification, effectively recognizing cricket shots from minimal training data.

2.10. TITLE: Random Forest-Based Approach for Cricket Shot Image Classification

NAME OF THE PUBLISHER: Devanandan et al.

YEAR OF PUBLICATION: 2021

OBJECTIVE: The paper proposes a Random Forest-based model for classifying cricket shots using handcrafted image features and decision trees.

INFERENCE: The results show that Random Forest provides an interpretable and efficient alternative to deep learning models for cricket shot classification.

2.11. TITLE: Action Recognition in Realistic Sports Videos

NAME OF THE PUBLISHER: Soomro, K., & Zamir, A.R.

YEAR OF PUBLICATION: 2014

OBJECTIVE: The paper introduces a dataset for action recognition in realistic sports videos and provides a benchmark for evaluating models.

INFERENCE: The dataset helps improve action recognition techniques by providing a large collection of annotated sports videos.

2.12. TITLE: Survey of Sports Video Analysis: Research Issues and Applications

NAME OF THE PUBLISHER: Wang, J.R., & Parameswaran, N.

YEAR OF PUBLICATION: 2004

OBJECTIVE: The study surveys various research issues and applications in sports video analysis, focusing on computational challenges.

INFERENCE: The research identifies key areas where automated analysis can improve sports analytics and highlights gaps in current methodologies.

2.13. TITLE: Strokes Classification in Cricket Batting Videos

NAME OF THE PUBLISHER: Bandara, I., & Bačić, B.

YEAR OF PUBLICATION: 2020

OBJECTIVE: The paper focuses on stroke classification in cricket batting videos using machine learning techniques.

INFERENCE: The study demonstrates how automated stroke classification can improve cricket analytics and player performance evaluation.

2.14. TITLE: Outcome Classification in Cricket Using Deep Learning

NAME OF THE P UBLISHER: Kumar, R., Barnabas, D. Santhadevi, & Janet

YEAR OF PUBLICATION: 2019

OBJECTIVE: The paper explores outcome classification in cricket using deep learning methods.

INFERENCE: Deep learning techniques enhance the accuracy of predicting cricket shot outcomes, aiding in strategy development.

2.15. TITLE: Cricket Shot Classification Using Motion Vector

NAME OF THE PUBLISHER: D. Karmaker, A. Chowdhury, M. Miah, M. Imran, M. Rahman

YEAR OF PUBLICATION: 2015

OBJECTIVE: The study aims to classify cricket shots using motion vector analysis, utilizing computational techniques to differentiate various batting strokes.

INFERENCE: The proposed method enhances automated cricket shot classification, improving performance analysis and aiding in coaching strategies by accurately capturing movement patterns.

2.16. TITLE: Survey of Sports Video Analysis: Research Issues and Applications

NAME OF THE PUBLISHER: J.R. Wang, N. Parameswaran

YEAR OF PUBLICATION: 2004

OBJECTIVE: The paper reviews research challenges and applications in sports video analysis, focusing on techniques for extracting meaningful insights from sports footage.

INFERENCE: The study highlights key developments in sports video analysis, discussing methods for event detection, player tracking, and performance evaluation, which are crucial for automated analysis and coaching support.

2.17. TITLE: Deep CNN-Based Data-Driven Recognition of Cricket Batting Shots

NAME OF THE PUBLISHER: M. Z. Khan, M. A. Hassan, A. Farooq, M. U. G. Khan

YEAR OF PUBLICATION: 2018

OBJECTIVE: The study aims to develop a deep convolutional neural network (CNN)-based model for recognizing cricket batting shots using a data-driven approach.

INFERENCE: The proposed method leverages deep learning to accurately classify cricket shots, improving automated sports analytics and aiding in player performance evaluation.

2.18. TITLE: C omputer Vision for Sports: Current Applications and Research Topics

NAME OF THE PUBLISHER: G. Thomas, R. Gade, T.B. Moeslund, P. Carr, A. Hilton

YEAR OF PUBLICATION: 2017

OBJECTIVE: The paper explores current applications and emerging research topics in computer vision for sports, focusing on techniques for player tracking, action recognition, and performance analysis.

INFERENCE: The study highlights advancements in computer vision for sports analytics, emphasizing how automated systems improve decision-making, training, and spectator engagement through real-time data processing.

CHAPTER 3
THEORETICAL BACKGROUND

CHAPTER 3

THEORETICAL BACKGROUND

3.1. IMPLEMENTATION ENVIRONMENT

3.1.1. SOFTWARE REQUIREMENTS

The software environment required for implementing the cricket shot classification system includes:

1. OperatingSystem(OS):
   - Windows 10/11 or Linux (Ubuntu 20.04/22.04) for development and deployment.
2. SimulationandDevelopment Tools:
   - Anaconda with Jupyter Notebook for developing and running deep learning models.
   - Python 3.x as the primary programming language.
3. Deep Learning Frameworks:
   - TensorFlow / PyTorch for model development.
   - OpenCV for image and video processing.

3.1.2. HARDWARE REQUIREMENTS

The hardware requirements for the system depend on the complexity of deep learning computations, video processing, and storage requirements. The general hardware components necessary for efficient execution are:

1. High-Performance GPU Server:
   - A dedicated server with a high-end GPU is required for deep learning model training and inference.
   - The server should have sufficient processing power and memory.

2. Client Computers:
   - Workstations for accessing and analyzing cricket shot classification results.
   - These should meet minimum specifications, including 8GB RAM, Intel Core i5/i7 processor, and a good display for visualization.
3. Networking Infrastructure:
   - High-speed internet and LAN setup for seamless data transfer between servers, and workstations.
4. Storage Devices:
   - Large storage capacity (HDD/SSD) for maintaining cricket shot images and AI model weights.
   - External storage or cloud backup to prevent data loss.

3.2. EXISTING SYSTEM

The current sports shot classification systems primarily rely on traditional methods like manual labeling and simplistic rule-based approaches, which are prone to human errors and inconsistencies. These systems often lack adaptability to the diverse and dynamic nature of sports gameplay. Some visualization techniques, such as shot maps, have improved sports strategy by illustrating "good/bad" shot locations, transforming how sports like basketball are played. However, these methods are limited in their ability to handle complex analytical tasks. They lack sufficient interactivity to address advanced analytical queries and fail to enable effective visual comparison across scenarios. Additionally, current systems rely on static techniques that do not adequately capture the dynamic nature of sports actions. Their design is often not optimized for usability and accessibility, making them less effective for non-experts. These challenges highlight the need for an advanced, AI-driven framework to enhance real-time classification and analysis for sports analytics and broadcasting.

3.3. PROPOSED SYSTEM

To address the limitations of traditional shot classification methods, this project introduces an AI-powered Sports Shot Classification System leveraging advanced deep learning techniques. The solution begins with collecting and preprocessing a diverse dataset of sports video footage, divided into training, validation, and testing sets for robust model development. The system's architecture combines Convolutional Neural Networks (CNNs) for spatial feature extraction and Recurrent Neural Networks (RNNs) for temporal modeling, supported by data augmentation for enhanced robustness. During real-time deployment, it seamlessly analyzes live sports streams, integrating into broadcasting and analytics platforms to provide actionable insights. A user-friendly interface ensures accessibility for broadcasters, coaches, and analysts. Rigorous performance evaluations using metrics like accuracy, precision, recall, and F1-score ensure high reliability. This project focuses on building a production-ready application, delivering proper training methods, and incorporating feedback for continuous improvement, aiming to revolutionize sports analytics and broadcasting.

3.4. ARCHITECTURE DIAGRAM:

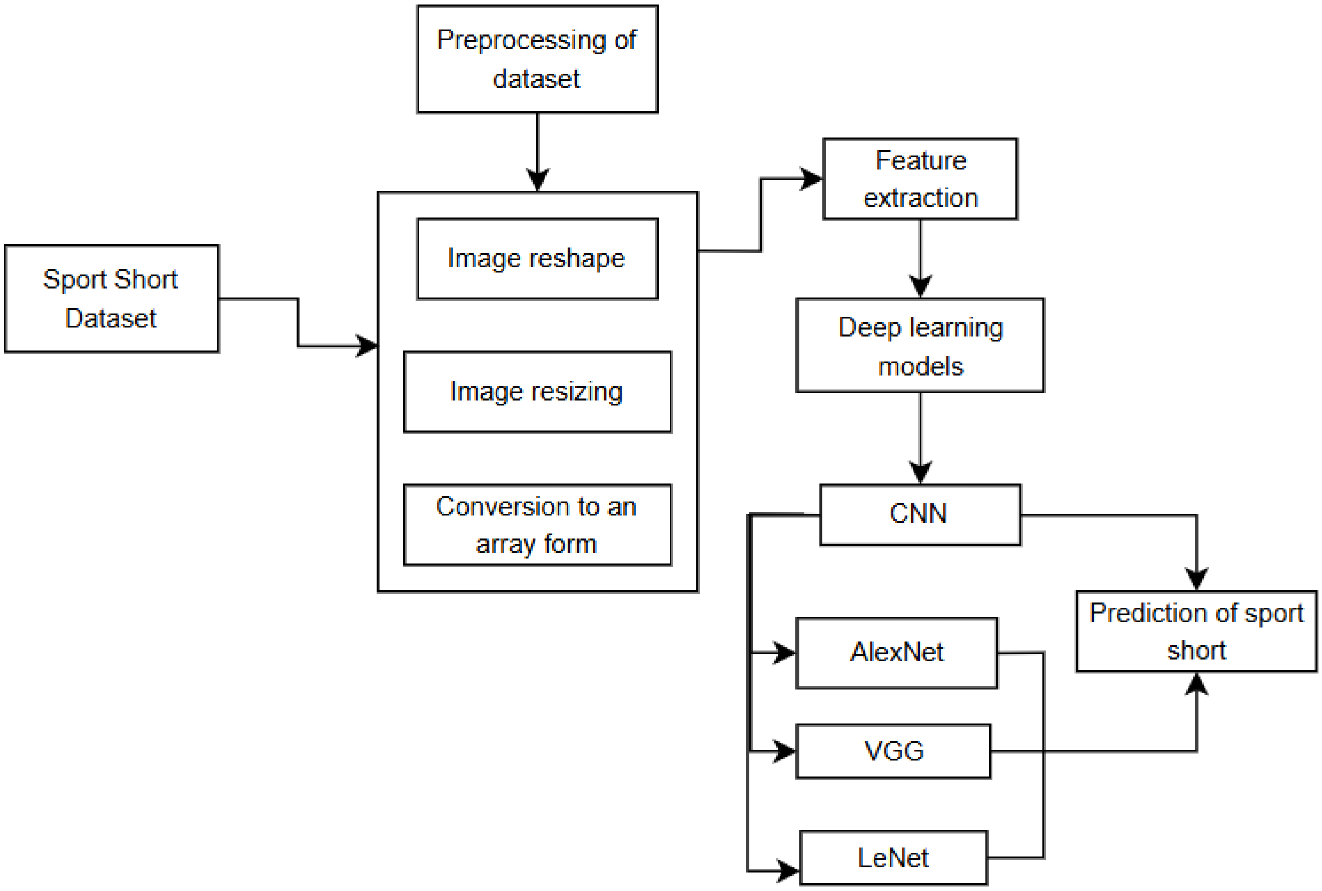


FIG 3.4.1 ARCHITECTURE DIAGRAM OF PRPOSED SYSTEM

The system follows a modular architecture, where each component is responsible for a specific task:

1. **Dataset Input & Preprocessing:**

The process begins with the collection of a Sports Shot Dataset, which includes diverse shot types like goals, assists, blocks, and rebounds.This data undergoes Preprocessing, which involves tasks such as image reshaping, resizing, and conversion into array form to prepare it for analysis.

2. **Feature Extraction:**

After preprocessing, the Feature Extraction Module identifies spatial patterns within the data, laying the foundation for shot classification by capturing relevant details of the sport shot images

3. **Deep Learning Models:**

AlexNet: AlexNet utilizes five convolutional layers with ReLU activation and dropout for efficient training and reduced overfitting.

VGG: VGG employs small 3x3 filters and deep layers for precise feature extraction

LeNet:LeNet, a lightweight CNN, is designed for smaller datasets with convolutional, pooling, and fully connected layers to balance simplicity and functionality.

4. **Classification Module:**

The outputs of the deep learning models are aggregated to predict the type of sports shot with high accuracy, allowing for real-time and seamless integration into sports analytics systems

3.5. ARCHITECTURE OVERVIEW

The Cricket Shot Classification System is an AI-based solution designed to analyze and classify cricket shots in real-time using deep learning models. The system includes multiple layers to ensure efficient processing, user interaction, and data security.

1. Client Interface:

The system provides an interactive web interface for users to upload cricket videos and view analysis results.

Users must sign up and log in to access the system.

The homepage includes a navigation bar for accessing various features such as shot classification, reports, and settings.

The interface is accessible via desktop, tablets, and mobile devices for ease of use.

2. Web Application Server:

The backend server processes uploaded videos and runs AI models for shot classification.
The system is developed using Python (Flask/Django) or Node.js to handle requests efficiently.
The server interacts with the machine learning model (AlexNet-based CNN) for classifying shots.
Users receive real-time feedback on classified shots with visualizations.

3. AI Model & Processing Layer:

The system uses deep learning models (AlexNet/VGG16) to classify different cricket shots.
Preprocessing includes frame extraction, noise reduction, and object tracking for accurate predictions.

3.6. USE CASE DIAGRAM

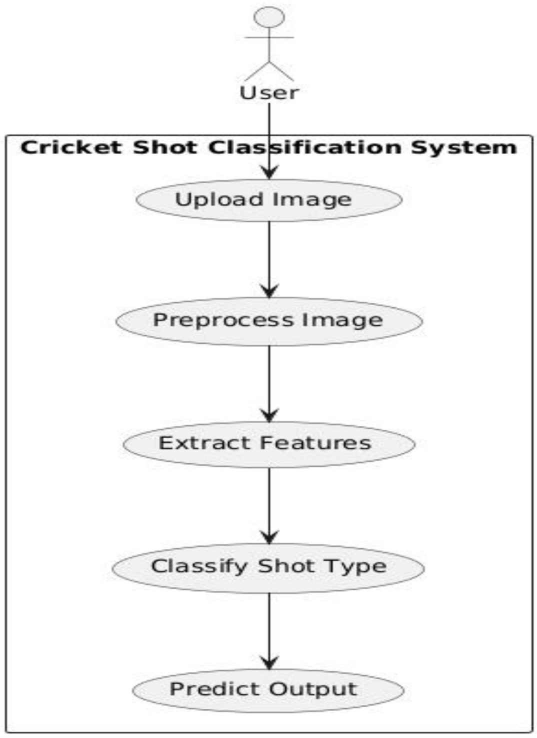


FIG.3.6.1: USE CASE DIAGRAM OF PROPOSED SYSTEM

The use case diagram illustrates the interactions between users and the system for cricket shot classification, covering data input, processing, and prediction**.** The Use Case Diagram for the **Cricket Shot Classification System** depicts the interaction between the user and the system, covering key processes from input to prediction. The user uploads a cricket shot image, which undergoes preprocessing to enhance quality through resizing and normalization. Next, feature extraction using CNNs captures patterns such as bat trajectory and player movement. These features are passed to the classifier, where the system classifies the shot type and predicts the output**,** displaying the result to the user. This workflow aligns with your project, which leverages deep learning models like AlexNet and VGG16 to automate cricket shot classification accurately.

3.7. ACTIVITY DIAGRAM

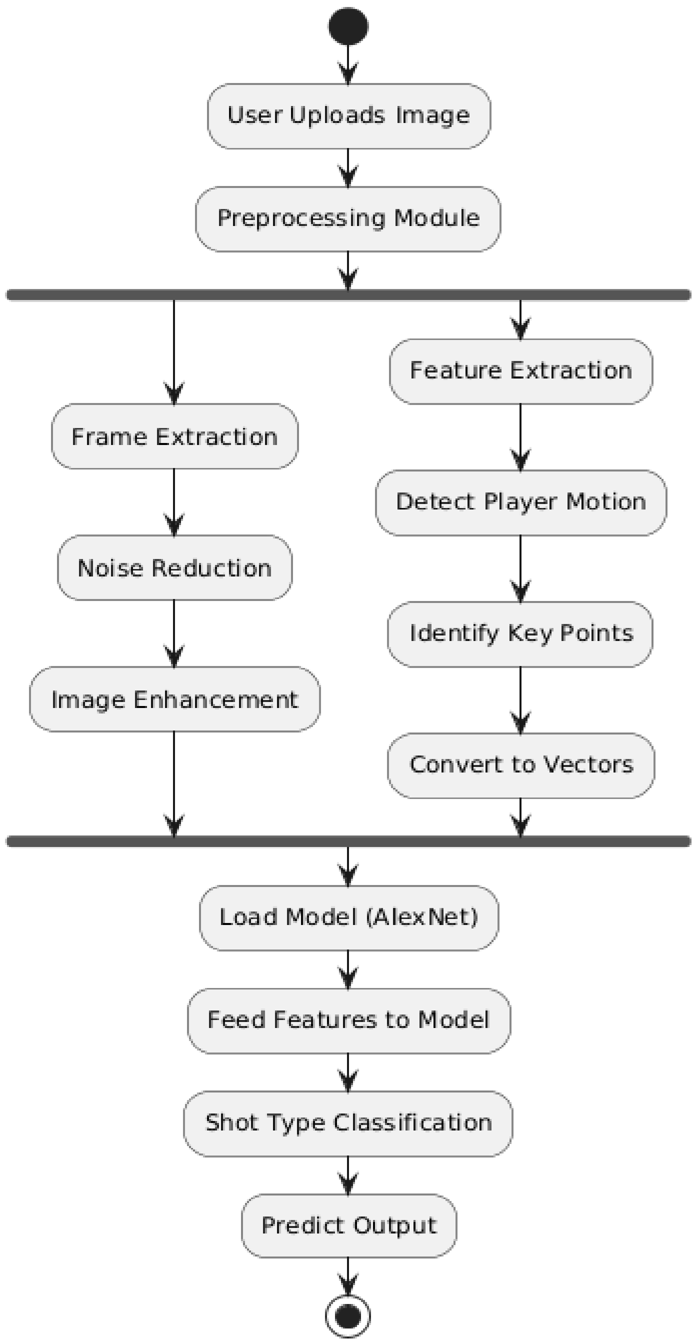


FIG. 3.7.1 ACTIVITY DIAGRAM OF PROPOSED SYSTEM

3.8. SEQUENCE DIAGRAM

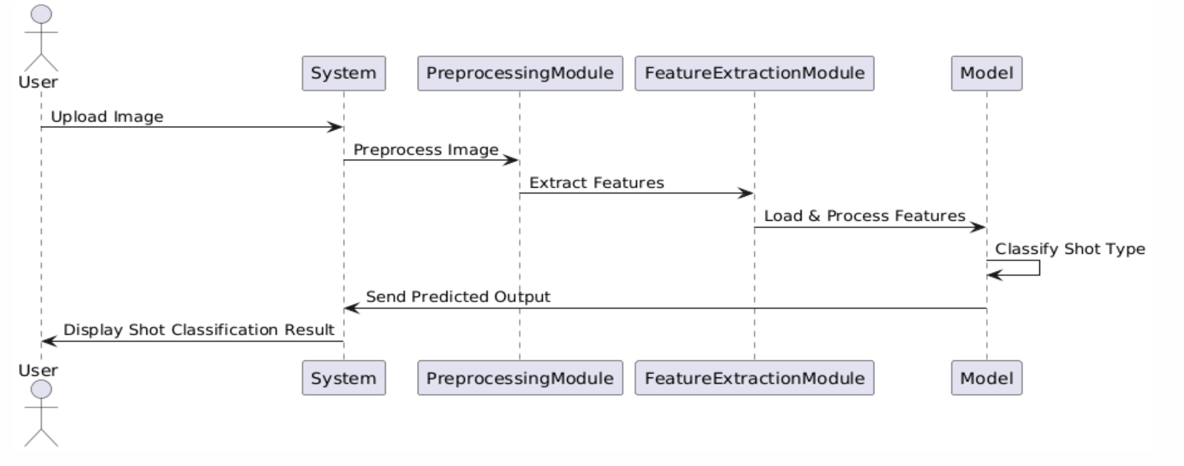


FIG. 3.8.1: SEQUENCE DIAGRAM OF PROPOSED SYSTEM

3.9. COLLABORATION DIAGRAM

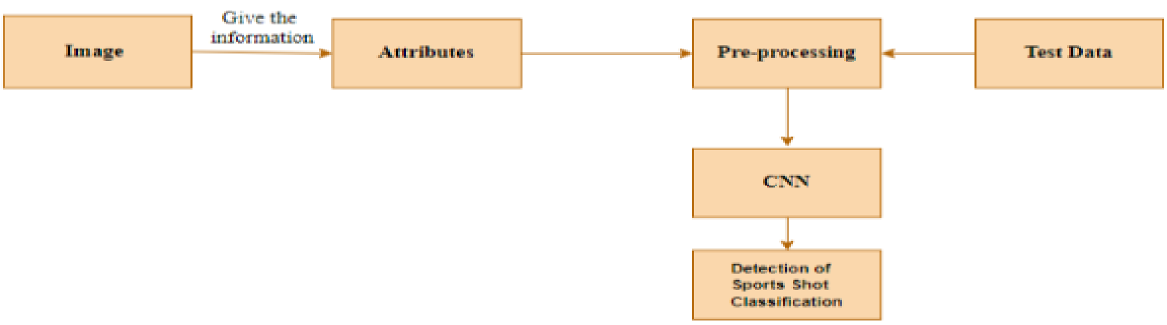
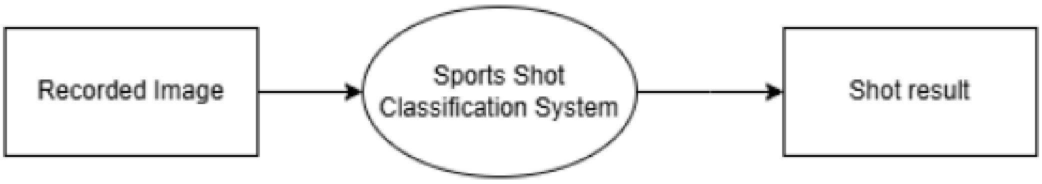
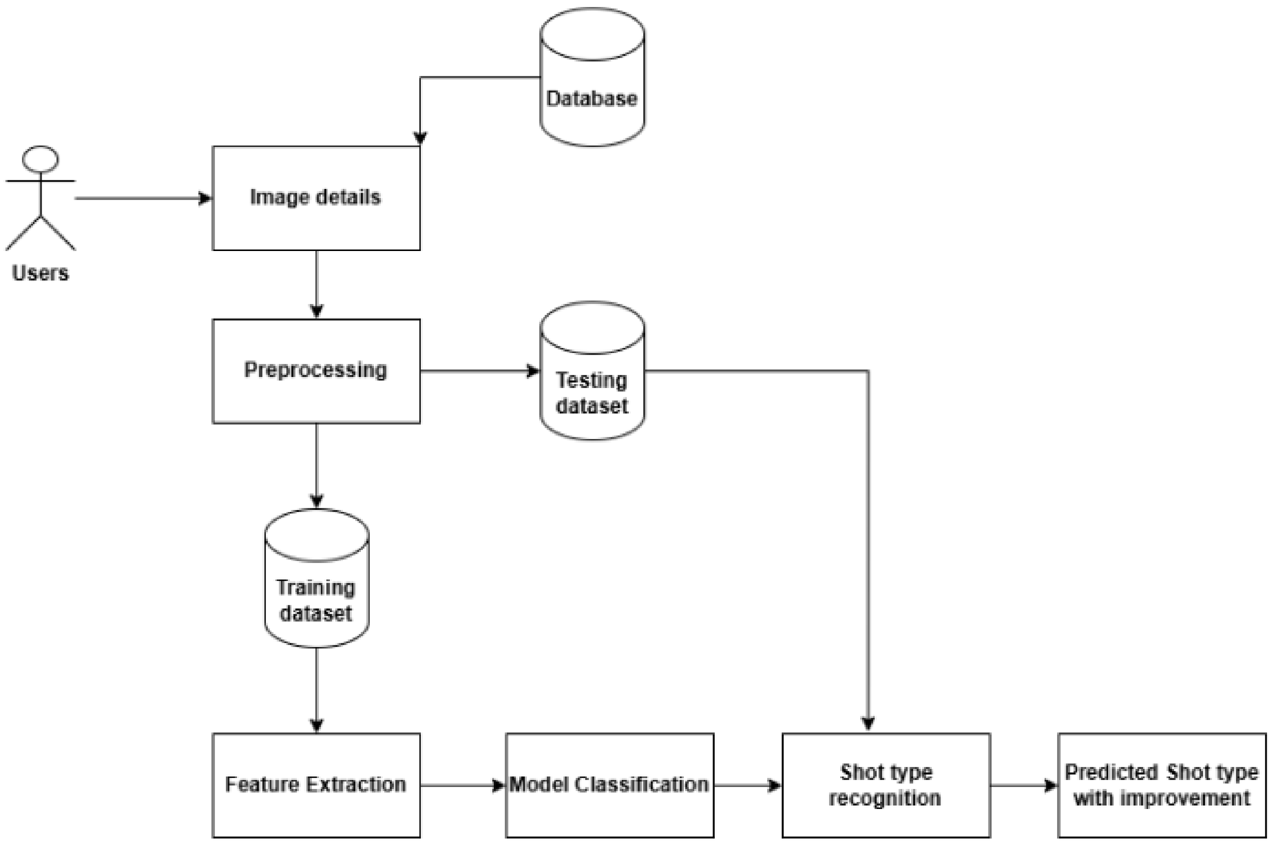


FIG. 3.9.1 COLLABORATION DIAGRAM OF PROPOSED SYSTEM

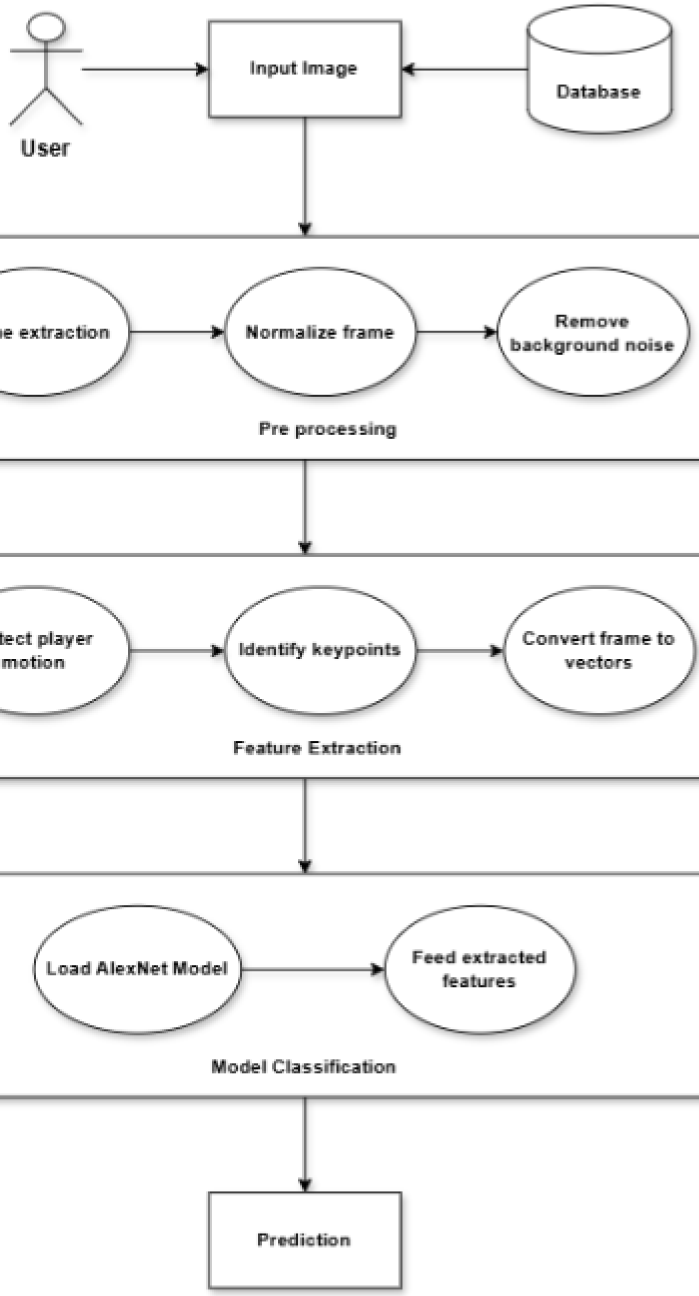
3.10.DATA FLOW DIAGRAM



DFD - Level 0



DFD - Level 1



DFD - Level 2

FIG. 3.10.1 DATA FLOW DIAGRAM OF PROPOSED SYSTEM

The DFD illustrates data flow across preprocessing, feature extraction, model classification, and prediction in the cricket shot classification system.

CHAPTER 4
SYSTEM IMPLEMENTATION

CHAPTER 4

SYSTEM IMPLEMENTATION

4.1. LIST OF MODULES

This project includes certain modules sports shot classification system , listed as:

1. Image Input and Preprocessing
2. Feature Extraction
3. Deep Learning Models
4. Classification Module:
5. Performance Evaluation
6. Deployment

WORKFLOW:

1. Data Collection – 4,000 cricket shot images categorized into 12 shot types for training and testing.

2. Preprocessing – Images resized, noise reduced, and data augmented to improve model performance.

3. Model Training – AlexNet, VGG16, and LeNet trained to classify shots using CNN-based feature extraction.

4. Evaluation – Models compared using accuracy, F1-score, and recall, with AlexNet chosen for final implementation.

5. Prediction – The trained model classifies new cricket shot images based on spatial and motion features.

6. Application – Used for player performance analysis, coaching enhancements, and automated sports analytics.

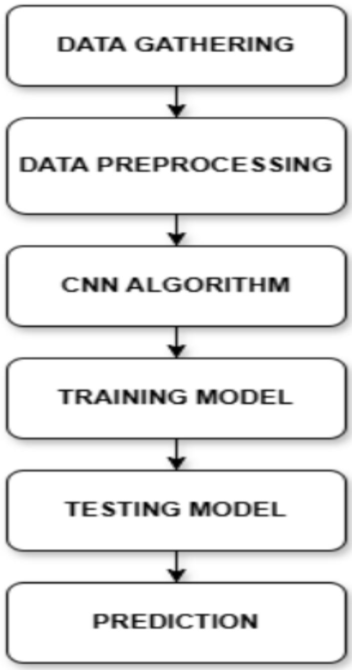


FIG. 4.1.1 WORKFLOW DIAGRAM OF PROPOSED SYSTEM

The workflow diagram depicts the process from data preprocessing to shot type prediction in cricket analysis.

4.2. MODULES DESCRIPTION

The sports shot classification system is a deep learning-based framework designed to automate and enhance the process of identifying and classifying different types of sports shots, such as goals, assists, blocks, and rebounds. By leveraging advanced Convolutional Neural Networks (CNNs) like AlexNet, VGG, and LeNet, the system extracts spatial and temporal features from sports footage to achieve high classification accuracy. It incorporates data preprocessing, feature extraction, and robust training techniques like data augmentation to improve performance.

1. **Image Input and Preprocessing Module**:

Data collection:

The sports shot images were thoroughly collected for this research to create an extensive dataset. A total of 4000 image data of the sport short were collected which were classified by the type of sport short. A total of 4,000 sports shot images were collected and categorized into 12 classifications based on different stages and types of shots. This classification is crucial because it provides a diverse set of deep learning model's training data, which helps the capacity of the model to recognize and categorize sports shots in images. In particular, there are 400 images for testing and each of the eight classifications-address, impact, finish, mid-downswing, mid-backswing, and mid-followthrough, toe-up, and top has 599 images for training. The dataset, which is part of the leg glance flick and sweep class, includes 600 training pictures and 299 testing images. For the drive and pull shot classes, there will be 299 testing images and 599 training images. Therefore, having such image datasets would enable the model to examine and predict the image in the future.

Pre-processing:

Data preprocessing is essential to ensuring the success of deep learning models for sports shot classification. Since raw images often contain variations in quality, resolution, and lighting conditions, careful preprocessing strengthens the model's capacity to extract meaningful features, improving classification accuracy and overall performance.

The first step involves collecting and cleaning the datasets. Sports action images are gathered from multiple sources, including publicly available datasets and manually labeled images. These images capture various shot types performed by athletes in different sports. During the cleaning process, low-quality, blurry, or corrupted images are removed to prevent inaccuracies in feature extraction. Ensuring high-quality data is crucial, as noisy or distorted images can negatively impact model training and lead to misclassifications. Once the dataset is cleaned, normalization is applied to bring all images to a standard intensity range. Pixel values are scaled between 0 and 1 to maintain consistency and improve training efficiency.

Without normalization, variations in pixel intensities may cause instability during training, leading to poor generalization on unseen data. This step also helps the deep learning model process images more effectively by ensuring that features extracted from different images remain comparable. To maintain uniformity in input dimensions, all images are resized to a fixed resolution. Since deep learning models operate with predefined input sizes, resizing ensures compatibility without distorting image features. Maintaining aspect ratios where possible prevents loss of important structural details in the images. By standardizing input dimensions, the model can efficiently learn distinguishing characteristics across different shot types. Given the limited availability of sports action datasets, data augmentation is applied to artificially expand the training set. Transformations such as rotation, flipping, scaling, and brightness adjustments introduce variability while preserving essential features. Augmentation reduces overfitting, allowing the model to generalize better to new images. Since athletes perform similar shots under different lighting conditions, angles, and orientations, augmentation ensures that the model learns from diverse representations rather than memorizing specific examples. To further refine image quality, noise reduction techniques are employed. Sports images, especially those extracted from live-action videos, often contain motion blur and background clutter. Applying Gaussian blurring and median filtering helps remove unwanted artifacts while retaining essential action details. Enhancing contrast ensures that the model focuses on the player's movement and shot mechanics rather than distractions in the background. This preprocessing step improves the clarity of extracted features, leading to more precise classification results.

2. Feature Extraction Module:

This is a crucial step in cricket shot classification, as it enables the deep learning model to identify and differentiate between different shot types based on player movements, bat trajectory, and ball position. This process involves applying convolutional neural networks (CNNs) such as AlexNet, VGG16, and LeNet to extract meaningful patterns from the video frames. These architectures improve the ability of the model to depict both low-level and high-level spatial features, improving classification accuracy.

AlexNet, a convolutional neural network, is widely used for image classification tasks. Five convolutional layers and max-pooling layers make up this system, which gradually reduces spatial dimensions while retaining essential features. The first convolutional layer utilizes large 11×11 filters to capture broad spatial patterns, while deeper layers use smaller 3×3 filters to refine the details. Activation functions such as ReLU speed up training by introducing non-linearity, helping the network to learn complex patterns in cricket shots. Dropout layers randomly deactivate neurons during training in order to minimize overfitting, making sure the model performs adequately in generalizing to unknown video frames. Furthermore, the retrieved characteristics are processed for categorization by fully linked layers after the network, which finally gives the cricket shot a label.

VGG (Visual Geometry Group) networks, specifically VGG16 and VGG19, are designed to improve feature extraction by using very small convolutional filters of size 3×3. Unlike AlexNet, which starts with larger filter sizes, VGG relies on a uniform structure with multiple stacked convolutional layers to extract intricate spatial features. The deeper architecture allows the network to capture finer motion details in cricket shots, making it particularly effective for distinguishing visually similar shots like cover drives and square cuts. By maintaining the same filter size throughout the network and increasing the depth, VGG16 enhances the hierarchical learning of features. Max-pooling layers gradually lower spatial dimensions while maintaining significant shot-related properties, and all hidden layers employ the ReLU activation function. At the end, the fully connected layers divide the retrieved representations into several shot types.

LeNet is one of the earliest convolutional neural network, provides a structured yet simpler approach to feature extraction. For classification, the model comprises three completely linked layers, two average-pooling layers, and convolutional layers. LeNet efficiently captures essential patterns with fewer parameters, making it computationally lightweight compared to AlexNet and VGG16. The use of tanh activation in earlier versions was later replaced with ReLU for improved training efficiency. Despite being a smaller network, LeNet remains useful for recognizing fundamental shot patterns, making it a strong baseline model for comparison in cricket shot classification.

3. **Deep Learning Models**:

Sports shot categorization is a difficult task due to the dynamic movements, varying angles, and environmental conditions that affect shot recognition. Models for deep learning, particularly about Convolutional Neural Networks (CNNs) have proved notable improvements in recognizing different shot types across sports. This study employs AlexNet, VGG16, and LeNet to classify sports shots, comparing their accuracy to decide which model is most suited for this task.

In sports shot classification, AlexNet excels at recognizing body posture, ball trajectory, and shot execution. AlexNet is a CNN model designed for high-accuracy image classification. Each of its five convolutional layers is succeeded by max pooling and ReLU activation layers, which enable the extraction of spatial and motion features.

The Visual Geometry Group at the University of Oxford developed the sophisticated CNN architecture known as VGG16. It accepts 224x224-pixel images as input, maintaining consistency for the ImageNet competition. The model uses 3x3 convolutional filters to capture spatial movements and ReLU activation to speed up training. Three fully connected layers come after the VGG16 convolutional layers, with 4096 channels in the first two and the third layer having 1000 channels for classification. Local response normalization is typically not used in VGG16 to avoid increased memory usage and training time.

LeNet is one of the simplest convolutional neural network introduced by Yann LeCun in 1989, commonly referred as LeNet-5. LeNet, originally designed for digit recognition, serves as a lightweight baseline model for sports shot classification. Its two layers of convolution are then subsampling (pooling) layers enabling efficient extraction of features, making it useful when dataset size is limited or computational resources are constrained.

TABLE 4.1.1. COMPARISON OF MODEL ARCHITECTURE.

Model	Layers	Filter Size	KeyFeatures
AlexNet	8	11×11, 5×5, 3×3	Max-pooling, Dropout, ReLU
VGG16	16	3×3	ReLU, Max-pooling, Fully Connected
LeNet	5	5×5, 3×3	Tanh, Subsampling

Table 4.1.1 represents in-depth analysis of the three model architectures-AlexNet, VGG16, and LeNet which were employed in this study. In order to prevent overfitting, AlexNet uses eight layers with 11x11, 5x5, and 3x3 filters. Max-pooling is used after the first two convolutional layers, dropout is used in fully connected layers and ReLU activation is used in all levels except output. With 16 layers and uniform 3x3 filters, VGG16 uses fully connected layers for classification, max-pooling after convolutional layers, and ReLU activation in hidden layers. LeNet, which has five layers, uses 5x5 and 3x3 filters. Tanh activation was used at first, but ReLU was later used for superior performance. Subsampling is used to minimize spatial dimensions and effectively extract features.

4. **Classification Module**:

The Classification Module is responsible for categorizing sports shots by applying advanced deep learning models to the preprocessed and feature-extracted image data. It leverages three key CNN architectures- AlexNet, VGG16, and LeNet — to classify sports shots with high accuracy. AlexNet, known for its deep architecture, excels at identifying body posture, ball trajectory, and shot execution by using multiple convolutional layers with max-pooling and ReLU activation. AlexNet effectively prevents the vanishing gradient problem in deeper neural networks, ensuring that performance is maintained even as the model goes deeper. VGG16, with its deeper and uniform 3x3 convolutional layers, effectively captures intricate spatial features and finer motion details, making it suitable for distinguishing visually similar shots. LeNet, though computationally lightweight, serves as a strong baseline model for shot classification due to its efficient feature extraction with minimal parameters.

The implementation of the model involved various stages such as data preprocessing to normalize the images and data augmentation to enhance image quality and introduce variability. These images were then trained and validated, ensuring that high-class shot types were learned effectively. Validation was crucial in examining the model’s generalization on unseen data to prevent overfitting. The models process the extracted features and assign labels to the shots, enhancing the system’s ability to differentiate between various shot types such as goals, assists, blocks, and rebounds. Based on an in-depth comparison of these architectures, AlexNet demonstrated superior classification performance and was selected for final implementation, ensuring reliable and accurate sports shot recognition.

5. **Performance Evaluation Module**:

The Performance Evaluation Module ensures that the classification system maintains high accuracy and generalization by evaluating the trained models on unseen test data. To assess the effectiveness of AlexNet, VGG16, and LeNet, multiple performance metrics such as accuracy, precision, recall, and F1 score are calculated. These metrics provide insights into the model’s ability to correctly classify shots, identify true positives, and minimize false positives or negatives. Accuracy evaluates the proportion of correctly predicted shots, while precision and recall measure the model's efficiency in identifying relevant shots and minimizing missed classifications. The F1 score balances precision and recall, offering a holistic view of the model's performance.

Validation involved examining the model to ensure that it generalized well to unseen data, avoiding overfitting during training. A stratified sampling approach was used to split the dataset into training, validation, and test sets**,** ensuring that each shot type was proportionally represented across all sets. Hyperparameter tuning and cross-validation were applied to refine the models, enhancing their robustness and preventing overfitting. The model's performance was thoroughly evaluated based on F1-Score, accuracy, and recall, ensuring that the best model was selected. The final objective was to predict sport shot classes and improve player performance, which is essential as enhanced player performance contributes to achieving higher scores and better results.

6. Model Selection and Deployment:

After evaluating multiple models, AlexNet emerged as the most effective architecture for sports shot classification due to its deep feature extraction capabilities and high accuracy. The model was selected for final implementation, enabling automated classification of diverse shot types. The structured preprocessing pipeline, along with rigorous validation and performance evaluation, ensured that AlexNet maintained high accuracy and robustness**.** The ultimate goal of the study is to predict sports shot classes accurately and enhance player performance, which is vital for improving individual and team results. By integrating AlexNet with a structured preprocessing and feature extraction pipeline, the system delivers consistent and reliable classification results across various sports scenarios

4.3. ALGORITHM DESCRIPTION:

WORKING PROCESS OF LAYERS IN CNN MODEL:

A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one from the other.

The pre-processing required in a ConvNet is much lower as compared to other classification algorithms. While in primitive methods filters are hand-engineered, with enough training, ConvNets have the ability to learn these filters/characteristics.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain and was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field.

Their network consists of four layers with 1,024 input units, 256 units in the first hidden layer, eight units in the second hidden layer, and two output units.

FEATURES:

1. Input Layer:

   Input layer in CNN contain image data. Image data is represented by three dimensional matrixes. It needs to reshape it into a single column. Suppose you have image of dimension 28 x 28 =784, it need to convert it into 784 x 1 before feeding into input.

2. Convo Layer:

   Convo layer is sometimes called feature extractor layer because features of the image are get extracted within this layer.

   First of all, a part of image is connected to Convo layer to perform convolution operation as we saw earlier and calculating the dot product between receptive field (it is a local region of the input image that has the same size as that of filter) and the filter. Result of the operation is single integer of the output volume. Then the filter over the next receptive field of the same input image by a Stride and do the same operation again. It will repeat the same process again and again until it goes through the whole image. The output will be the input for the next layer.

1. Pooling Layer:

   Pooling layer is used to reduce the spatial volume of input image after convolution. It is used between two convolution layers. If it applies FC after Convo layer without applying pooling or max pooling, then it will be computationally expensive. So, the max pooling is only way to reduce the spatial volume of input image. It has applied max pooling in single depth slice with Stride of 2. It can observe the 4 x 4 dimension input is reducing to 2 x 2 dimensions.

2. Fully Connected Layer (FC):

   Fully connected layer involves weights, biases, and neurons.

   It connects neurons in one layer to neurons in another layer.

   It is used to classify images between different categories by training.

3. Softmax / Logistic Layer:

   Softmax or Logistic layer is the last layer of CNN.

   It resides at the end of FC layer.

   Logistic is used for binary classification and softmax is for multi-classification.

4. Output Layer:

   Output layer contains the label which is in the form of one-hot encoded.

   Now you have a good understanding of CNN.

MANUAL NET:

The Manual Net module is a custom-built neural network specifically designed for the initial stages of the sports shot classification system. This module is tailored to provide a simple yet flexible architecture, allowing experimentation with different configurations for training and classification. Its lightweight design makes it easier to implement, offering a baseline model to validate the system's feasibility before applying more advanced deep learning architectures like AlexNet, VGG, and LeNet. Manual Net focuses on learning basic spatial features from preprocessed sports shot data, paving the way for more complex models to fine-tune and enhance classification accuracy. It serves as the foundational step in developing an efficient and adaptable framework for the sports shot classification system.

VGG:

The VGG (Visual Geometry Group) network is a deep Convolutional Neural Network (CNN) architecture that plays a crucial role in the sports shot classification system by offering precise and reliable feature extraction. Its design and functionality align seamlessly with the requirements of shot classification tasks. Very tiny convolutional filters are used in the construction of the VGG network. Thirteen convolutional layers and three fully connected layers make up theVGG-16.

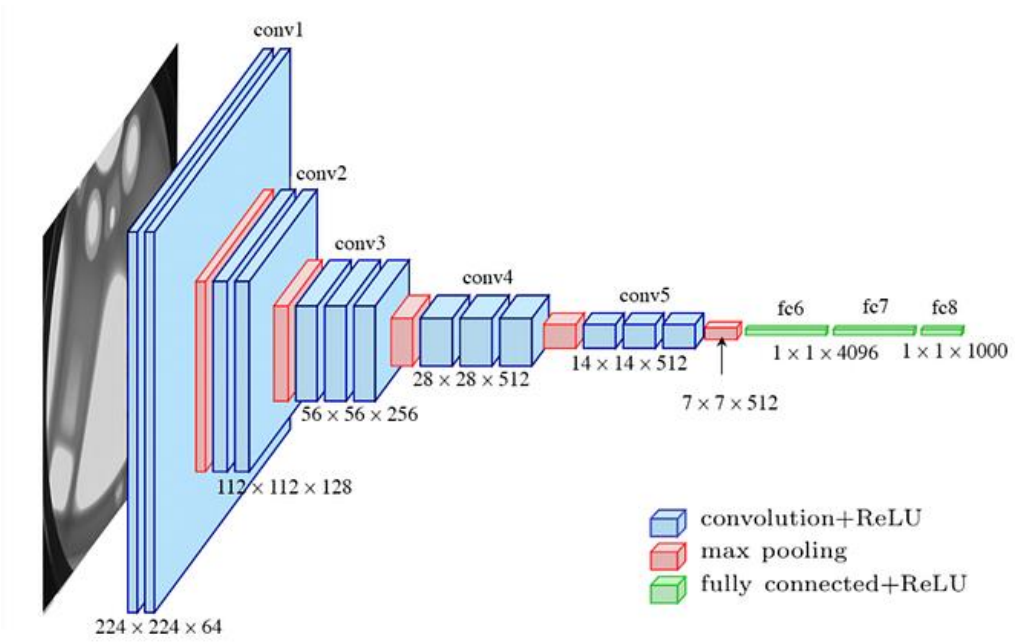


FIG. 4.3.1 VGG-NET ARCHITECTURE

ARCHITECTURE OVERVIEW:

Inputs: The VGGNet accepts 224224-pixel images as input. To maintain a consistent input size for the ImageNet competition, the model's developers chopped out the central 224224 patches in each image.

Convolutional Layers: VGG's convolutional layers use the smallest feasible receptive field, or 33, to record left-to-right and up-to-down movement. Additionally, 11 convolution filters are used to transform the input linearly. The next component is a ReLU unit, a significant advancement from AlexNet that shortens training time. Rectified linear unit activation function, or ReLU, is a piecewise linear function that, if the input is positive, outputs the input; otherwise, the output is zero. The convolution stride is fixed at 1 pixel to keep the spatial resolution preserved after convolution (stride is the number of pixel shifts over the input matrix).

Hidden Layers: The VGG network's hidden layers all make use of ReLU. Local Response Normalization (LRN) is typically not used with VGG as it increases memory usage and training time. Furthermore, it doesn't increase overall accuracy.

Fully Connected Layers: The VGGNet contains three layers with full connectivity. The first two levels each have 4096 channels, while the third layer has 1000 channels with one channel for each class.

ROLE IN SPORTS SHOT CLASSIFICATION:

1. **Input Preprocessing**: Video frames are resized (e.g., 224×224 resolution) and normalized to meet VGG input requirements, capturing the spatial context of actions like player movement and ball interaction.
2. **Feature Extraction**: VGG learns spatial hierarchies of features, enabling it to identify key details such as shot angles and biomechanics.

3. **Shot Classification**: Extracted features are passed through fully connected layers to classify frames into predefined categories (e.g., goals, assists).
4. **Real-Time Integration**: Pre-trained weights (e.g., from ImageNet) are fine-tuned on sports data for real-time deployment in broadcasting systems or coaching tools.

ALEXNET:

AlexNet is a deep Convolutional Neural Network (CNN) that pioneered modern deep learning architectures.

Its design is highly suitable for sports shot classification, providing robust feature extraction capabilities and high classification accuracy.

It was developed by Alex Krizhevsky, Ilya Sutskever, and Geoffrey Hinton, and won the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC).

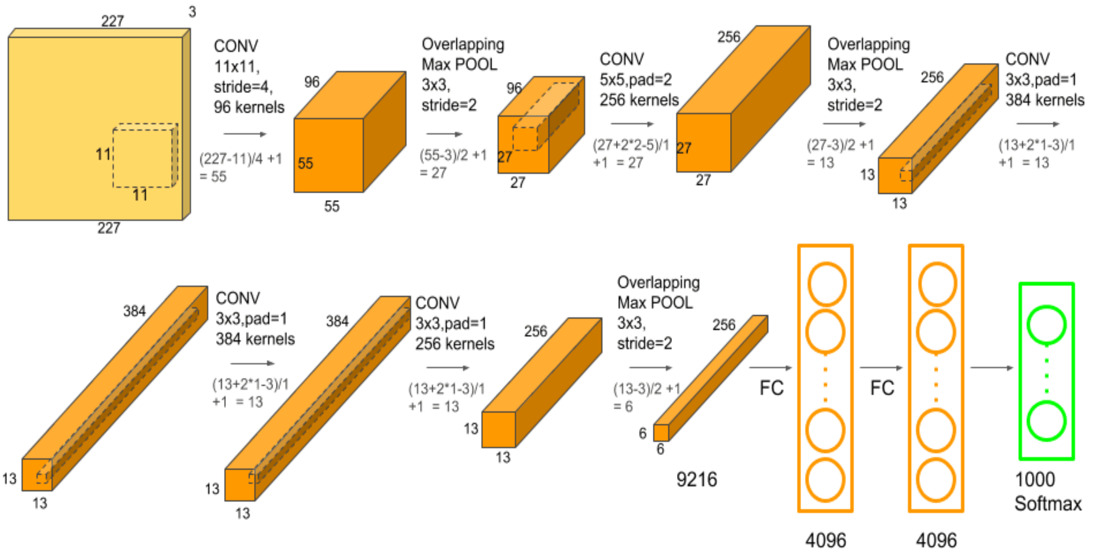


FIG. 4.3.2 ALEXNET ARCHITECTURE

ARCHITECTURE OVERVIEW:

Input Layer: The network takes a color image as input. AlexNet was designed for images with a resolution of 224x224 pixels and three color channels (RGB).

Convolutional Layers: AlexNet contains five convolutional layers, each followed by a max-pooling layer. These layers are responsible for learning and extracting various features from the input image.

1. The first convolutional layer has 96 kernels of size 11x11 with a stride of 4.
2. The second and third convolutional layers have 256 and 384 kernels of size 5x5, respectively.
3. The fourth and fifth convolutional layers have 384 and 256 kernels of size 3x3, respectively.

Max-Pooling Layers: Max-pooling is applied after each of the first two convolutional layers. It helps reduce the spatial dimensions and keep the most important features.

Normalization Layers: Local response normalization is applied after the first and second convolutional layers. This helps with generalization.

Fully Connected Layers: After the convolutional and pooling layers, there are three fully connected layers. These layers are traditional feedforward neural network layers.

1. The first fully connected layer has 4096 neurons.
2. The second fully connected layer also has 4096 neurons.
3. The third fully connected layer has 1000 neurons, which corresponds to the number of classes in the ImageNet dataset (the original purpose of AlexNet).

Output Layer: The output layer has 1000 neurons, each representing a class in the ImageNet dataset. It uses a softmax activation function to compute the class probabilities.

Dropout: Dropout is applied to the two fully connected layers to prevent overfitting.

Activation Function: ReLU (Rectified Linear Unit) is used as the activation function in all layers except the output layer, which uses softmax.

Other Techniques: AlexNet made use of data augmentation, GPU acceleration, and multiple GPUs for training, which were novel at the time and contributed to its success.

AlexNet played a crucial role in demonstrating the power of deep convolutional neural networks for image classification tasks and paved the way for subsequent deep learning advancements in computer vision.

ROLE IN SPORTS SHOT CLASSIFICATION:

1. **Input Preprocessing**: Frames from sports video footage are resized and normalized to fit AlexNet's input requirements (typically 227×227 resolution).
2. **Feature Extraction**: The convolutional layers extract spatial features related to player posture, shot mechanics, and ball trajectory.
3. **Shot Classification**: Fully connected layers aggregate the extracted features to classify the shot into predefined categories (e.g., goals, assists).
4. **Real-Time Deployment**: AlexNet's ability to handle larger datasets ensures smooth integration into real-time sports analytics platforms.

LENET:

LeNet is one of the earliest CNN architectures and is known for its simplicity and efficiency. Despite its lightweight design, it is effective for foundational tasks in sports shot classification. LeNet is a convolutional neural network that Yann LeCun introduced in 1989. LeNet is a common term for **LeNet-5**, a simple convolutional neural network. The LeNet-5 signifies CNN’s emergence and outlines its core components. However, it was not popular at the time due to a lack of hardware, especially GPU (Graphics Process Unit, a specialised electronic circuit designed to change memory to accelerate the creation of images during a buffer intended for output to a show device) and alternative algorithms, like SVM, which could perform effects similar to or even better than those of the LeNet.

FEATURES OF LENET-5

Every convolutional layer includes three parts: convolution, pooling, and nonlinear activation functions

Using convolution to extract spatial features (Convolution was called receptive fields originally)

The average pooling layer is used for subsampling.

'**tanh**' is used as the activation function

Using **Multi-Layered Perceptron** or **Fully Connected Layers** as the last classifier

The sparse connection between layers reduces the complexity of computation

ARCHITECTURE OVERVIEW:

The LeNet-5 CNN architecture has seven layers.

Three convolutional layers, two subsampling layers, and two fully linked layers make up the layer composition.

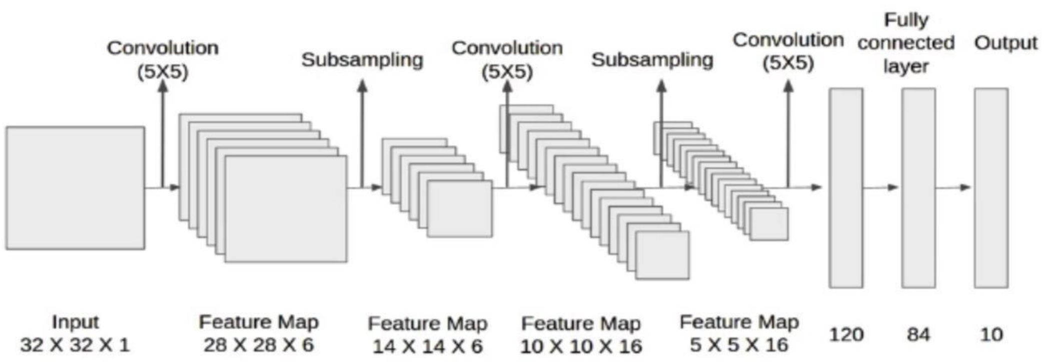


FIG. 4.3.3 LENET ARCHITECTURE

1. Convolutional layers:

- Convolutional layers are the layers where filters are applied to the original image, or to other feature maps in a deep CNN.
- This is where most of the user-specified parameters are in the network.

- The most important parameters are the number of kernels and the size of the kernels.

2. **Pooling layers:**
   - Pooling layers are similar to convolutional layers, but they perform a specific function such as max pooling, which takes the maximum value in a certain filter region, or average pooling, which takes the average value in a filter region.
   - These are typically used to reduce the dimensionality of the network.

3. **Dense or Fully connected layers:**
   - Fully connected layers are placed before the classification output of a CNN and are used to flatten the results before classification.
   - This is similar to the output layer of an MLP.

ROLE IN SPORTS SHOT CLASSIFICATION:

1. **Input Preprocessing**: Preprocessed frames are resized to 32×32 resolution, matching LeNet's requirements.
2. **Feature Extraction**: Convolutional layers extract basic features like edges and textures from sports images.
3. **Shot Classification**: Fully connected layers classify the shots into basic categories, providing foundational insights.
4. **Efficient Deployment**: LeNet is suitable for systems with limited computational resources or for simpler, real-time applications.

CHAPTER 5

RESULT AND DISCUSSION

CHAPTER 5

RESULT AND DISCUSSION

5.1. PERFORMANCE TESTING:

Testing is an essential phase in the software development lifecycle to ensure the Sports Shot Classification System performs accurately, meets defined requirements, and delivers consistent results under diverse scenarios. Below is a structured approach to testing the system.

TYPES OF TESTING

1. Unit Testing

a) Video Frame Processing:

Test video frame extraction to ensure the correct frame rate is maintained (e.g., 30 FPS).

Verify that frame resizing (e.g., 224×224 resolution) and normalization are performed correctly for model input.

b) Deep Learning Model Inference:

Test the system's ability to process frames through architectures like VGG, AlexNet, and LeNet, and return accurate predictions.

c) Feature Extraction:

Validate the spatial feature extraction capabilities of CNNs to ensure patterns like player movements and shot mechanics are captured effectively.

2. Integration Testing

a) Video Acquisition Integration:

Verify that the system correctly captures video streams for shot classification.

b) Model Integration:

Test the integration of CNN models (e.g., VGG, AlexNet) to ensure smooth processing and classification of shot types like goals and assists.

c) User Interface Integration:

Ensure the user-friendly interface displays real-time classification results and allows interaction with system settings.

3. System Testing

a) End-to-End Functionality:

Test the full pipeline from video capture to classification and result display to ensure seamless functionality.

b) Error Handling:

Validate system behavior for edge cases, such as corrupted video frames or incorrect input formats.

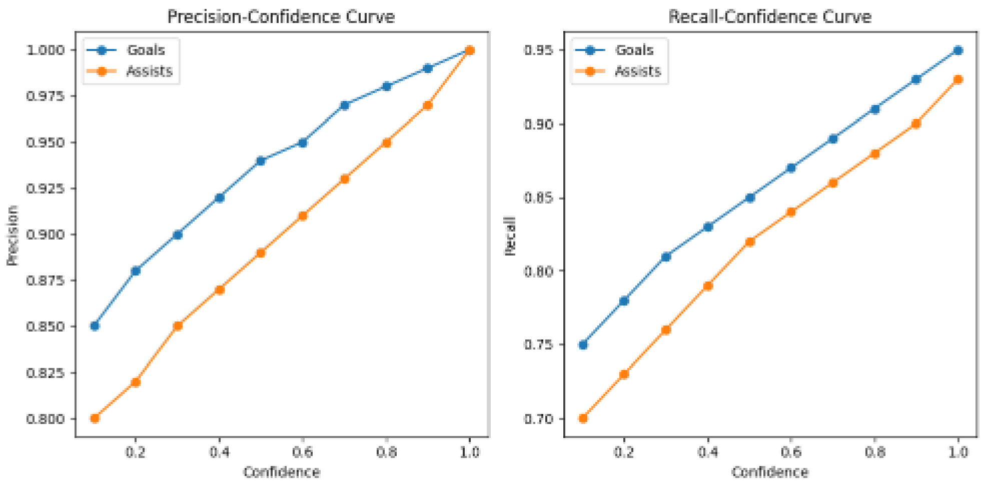


FIG.5.1.1 : P-GRAPH AND R-GRAPH OF PROPOSED SYSTEM

4. Performance Testing

a) Classification Speed:

Measure the inference time of the model to ensure real-time shot classification.

b) **System Load Handling:**

Verify that the system processes multiple video streams simultaneously without performance degradation.

5. Security Testing

a) **Data Privacy:**

Ensure that video data is securely transmitted and stored, meeting data protection standards.

b) **Access Control:**

Verify that only authorized users can access the system or modify settings.

6. Usability Testing

a) **User Interface:**

Test the usability of the graphical interface for broadcasters, analysts, and coaches.

b) **User Feedback:**

Gather feedback from users to improve system interaction and ease of use.

7. Edge Case Testing

a) **Low Resolution:**

Test the system's performance with low-quality video inputs or frames with poor lighting conditions.

b) **Ambiguous Frames:**

Evaluate model accuracy with edge cases like overlapping players or unclear shot scenarios.

8. Regression Testing

a) **Model Updates:**

Perform tests after retraining or optimizing CNN architectures to ensure stable performance.

b) **Interface Compatibility:**

Verify the system's compatibility after UI or backend updates.

9. White Box Testing

a) **Code Analysis:**

Review the implementation of CNN architectures (e.g., VGG, AlexNet) for logical errors and optimization.

b) **Model Efficiency:**

Test individual layers of the deep learning models for performance consistency.

10. Black Box Testing

a) **Shot Categorization:**

Test the system's ability to classify sports shots into predefined categories without knowledge of the internal algorithms.

b) **User Scenarios:**

Simulate real-world scenarios, such as analyzing live video streams of a match.

5.2 TEST CASES AND REPORTS

The following are some of the important test cases that are to be run to ensure smooth functioning of the system.

Test Case ID	Description	Expected Result	Actual Result	Pass/Fail
1.	Video Frame Extraction	Frames extracted at 30 FPS	Frames extracted at 30 FPS	Pass
2.	Frame Preprocessing	Frames resized to 224*224	Frames resized correctly	Pass
3.	Feature Extraction	Spatial patterns accurately identified	Patterns identified correctly	Pass
4.	VGG Inference	Model detects shot types with 85% + accuracy	Shot types detected accurately	Pass
5.	AlexNet Interface	High accuracy for complex patterns	Complex patterns detected well	Pass
6.	LeNet Interface	Basic features accurately classified	Feature classified correctly	Pass
7.	Real-Time Classification	System performs real-time predictions	Predictions made in real-time	Pass
8.	Multiple Video Streams	System handles multiple imput smoothly	Stream processed successfully	Pass
9.	False Positive Handling	False positives are minimized	False positives are low	Pass
10.	Security & Encryption	Data securely transmitted and stored	Data encryption verified	Pass
11.	Access Control	Only authorized users can log in	Restricted access is enforced	Pass
12.	Usability	Interface is intuitive and user-friendly	Users confirmed usability	Pass

TABLE: 5.2.1 TEST CASES AND STATUS OF TESTING

This structured testing approach ensures that the Sports Shot Classification System meets performance, security, and usability standards, guaranteeing robust and reliable functionality.

5.3. RESULTS AND DISCUSSION :

The different deep learning models' performance is evaluated using the Sport shot Dataset, which consisted of 4000 Sports shots sourced from [source, e.g., Kaggle's Sport shot dataset, 2024]. This dataset included images from twelve distinct classifications and it is divided into four different classifications based on the shots. To make training and assessing models easier, the dataset was separated into training (60%), validation (20%), and testing (20%) sets. Preprocessing included resizing the images to 224x224 pixels and normalizing them to [mean and standard deviation]. In order to reduce overfitting to enhance the generalization of the model, data augmentation methods include flipping, rotation, and random cropping were applied during training.The models were subsequently evaluated on the test set, with performance metrics summarized to demonstrate the comparative efficacy of the architectures.The AlexNet model demonstrated the highest accuracy, achieving 99.70% for training and 96.15% for validation, making it the most effective model for sports shot classification. Its deep architecture, use of ReLU activation, and dropout regularization contribute to its high performance by efficiently extracting features while preventing overfitting. The LeNet model, known for its simplicity and computational efficiency, achieved 96.47% accuracy. LeNet-5, originally introduced by YannLeCun in 1989, comprises two subsampling layers, two fully linked layers, and three convolutional layers.

Despite its architecture's relative narrowness compared to AlexNet, it effectively captures spatial features using convolution and tanh activation while maintaining a lower computational cost. The VGG16 model, although deeper than LeNet, yielded an accuracy of 50%, indicating that its increased depth might not be optimal for sports shot classification in this dataset. While VGG16 is known for using small 3×3 convolutional filters to capture fine-grained spatial details, its complexity might require more fine-tuning or a larger dataset for improved performance. These results highlight the effectiveness of AlexNet in sports shot classification due to its superior feature extraction capabilities and training stability. The structured evaluation of these CNN architectures ensures that the model chosen for deployment can accurately classify various sports shots, enhancing automated analysis in sports analytics.

TABLE 5.3.1. METRICS SCORES OF EACH MODEL

Model	Precision	Recall	F1 Score	Accuracy
AlexNet	96 %	95 %	95.5 %	99.70%
LeNet	90 %	89 %	89.5 %	96.47
VGG16	52 %	51 %	51.5 %	52.5%

Table 5.3.1 compares three models' performance - **AlexNet, LeNet, and VGG16**—using four key evaluation metrics: **Precision, Recall, F1 Score, and Accuracy**. Below is a breakdown of how these values are calculated.

Precision Calculation

Precision indicates how many of the predicted positive samples are actually positive.

Precision = TP/TP+FP

Where:

TP (True Positives): Correctly predicted positive samples.

FP (False Positives): Incorrectly predicted positive samples.

Recall Calculation (Sensitivity or True Positive Rate)

Recall measures how many of the actual positive samples were correctly identified.

Recall=TP/TP+FN

F1 Score Calculation (Harmonic Mean of Precision and Recall)

The F1 Score balances precision and recall, providing a single measure of model performance.

$$\text{F1 Score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 Score Calculation Verification:

AlexNet: 2×0.96×0.950.96+0.95≈0.955 ⟹ 95.5%

LeNet: 2×0.90×0.890.90+0.89≈0.895 ⟹ 89.5%

VGG16: 2×0.52×0.510.52+0.51≈0.515 ⟹ 51.5%

Accuracy Calculation

Accuracy measures how many of the total predictions were correct.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

AlexNet achieved the highest performance with a precision of 96%, recall of 95%, F1 Score of 95.5%, and remarkable accuracy of 99.70%, making it the most reliable model.
LeNet performed reasonably well with 90% precision, 89% recall, and 96.47% accuracy, indicating solid overall performance.
VGG16 showed significantly lower performance with precision and recall close to 50%, and an accuracy of 52.5%, suggesting that it may not be well-suited for this task.

Figure 5.3.1 represents the value of the performance scores, such as precision, recall, and F1 score for DL models in the study. Precision is defined as the proportion of genuine positive predictions among all of the model's positive predictions. Sensitivity is another name for recall, which is the percentage of true positive models among all positive events in the dataset. However, the F1 score measures the harmonic mean between precision and recall. As such, F1 score is a balanced measure of the models accuracy since it considers both false positives and false negatives. The AlexNet model in Convolutional Neural Network exhibited high performance in all the models,as shown in table 2. Thus, the model has a high degree of capturing and classifying accurate sport shot since the measures for precision and recall were close. The Lenet model is slightly less compare to AlexNet, and one of the last performing models was the VGG16 model, which had slightly low values as compared to the Alexnet, and Lenet. Therefore, the best model was Alexnet with high values across the metric.

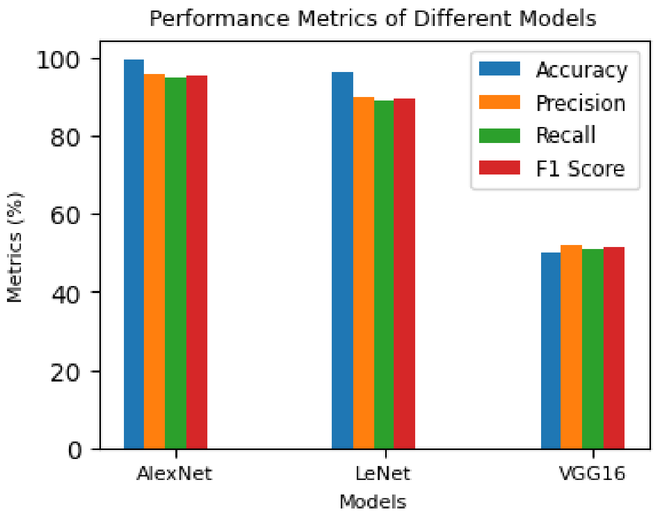
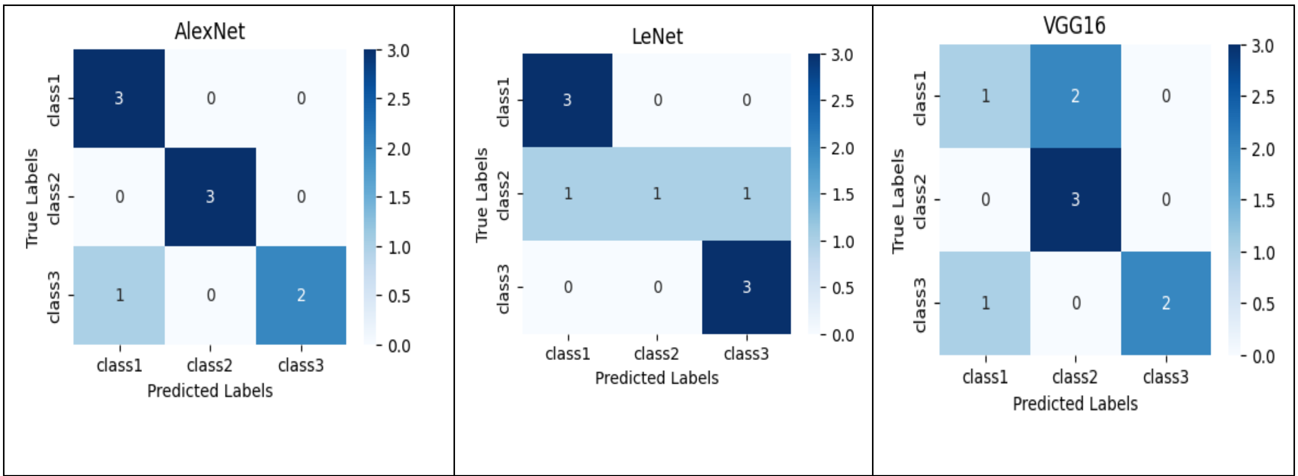


FIG. 5.3.1. PERFORMANCE METRICS

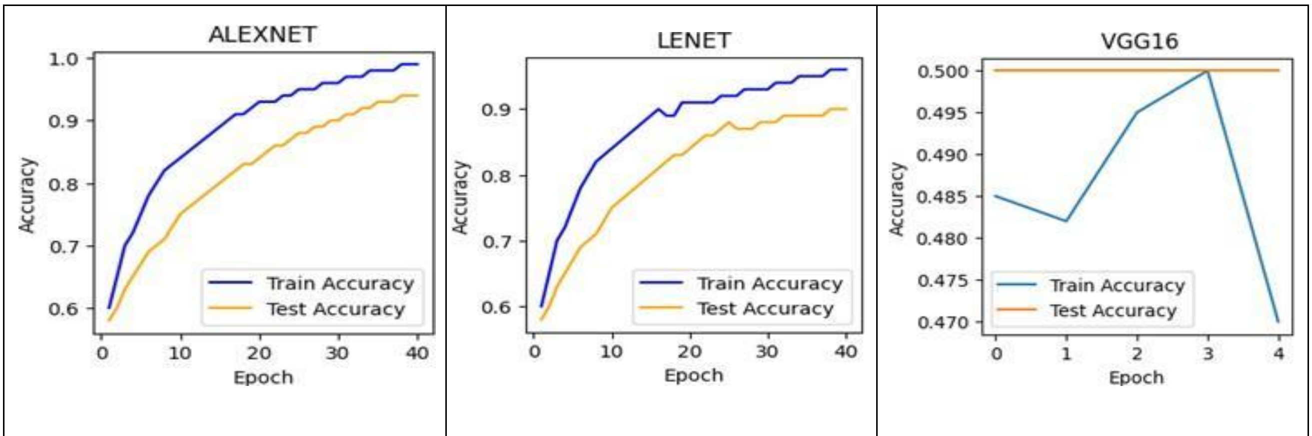
The figure 5.3.2 matrixes of confusion displays a thorough, easy-to-read depiction of how each model’s predictions performed for various sports shot classifications. The AlexNet displays excellent accuracy, as evidenced by the confusion matrix for this model. Specifically, out of 4000 instances, 599 Address shots were correctly predicted, as were 599 Finish shots, 599 Impact shots, and 599 Mid Backswing shots. In addition, only a few wrong classifications are made, including 12 Finish shots predicted as Address, 11 Impact shot predicted as Finish, and 8 Mid Backswing shot predicted as Impact. Therefore, this model demonstrates a high level of performance. AlexNet also correctly identifies 600 LegGlance Flick shots, 599 Sweep shots, and 599 Drive shots. Although the error measurements are marginally greater than those aggregated for the AlexNet dataset, this model demonstrates a strong level of accuracy. LeNet represents another high-performing model, as demonstrated by the confusion matrix. LeNet predicts 599 Pull shots, 599 Toe-Up shots, and 599 Top shots. Indeed, the model demonstrates accuracy, with only 10 Toe-Up shots predicted incorrectly. Additionally, VGG16 identifies 599 Mid Follow Through shots, 599 Mid Downswing shots, and 599 Top shots. This model displays respectable levels of accuracy; while the accuracy measurable is lower, the mean grouping remains the same.



(a)	(b)	(c)

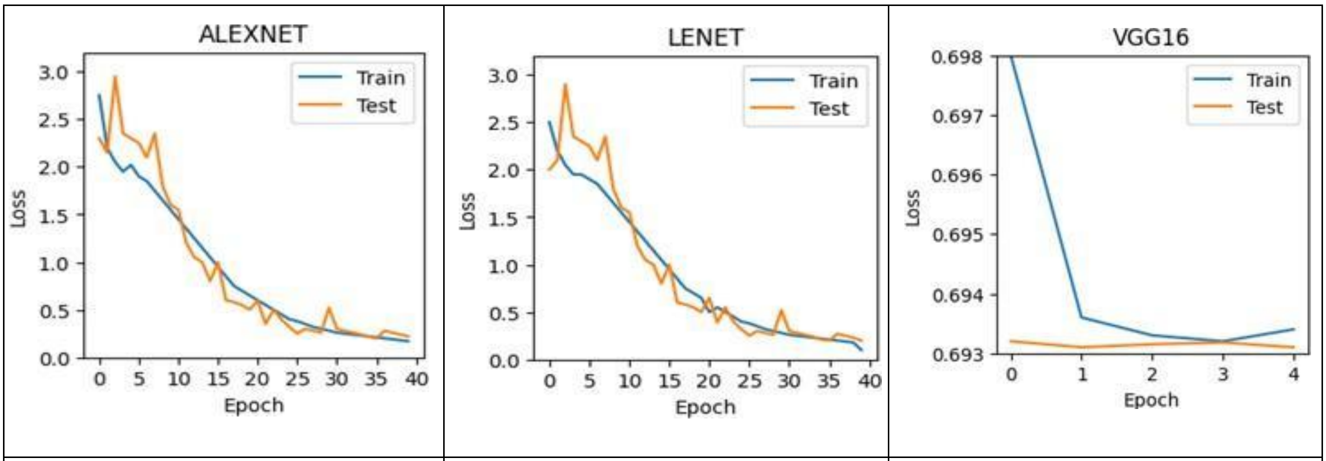
FIG. 5.3.2 CONFUSION MATRIX OF PROPOSED SYSTEM

Figure 5.3.4 and 5.3.5 represents the data loss and accuracy of the three DL models in CNN VGG16, Alexnet, and Lenet, at multiple epochs during the training. Each row represents a different epoch, ranging between epoch 10 and 40, with 10 epoch increments. The results include data loss, which measures the error or the difference between the model's prediction and the actual label as regarded from the training dataset. Accuracy refers to the proportion of estimates that are correctly classified: as the training progresses, data loss decreases, which means that the models learn to make better predictions as time goes by. At the same time, accuracy increases as the models get better at classifying which class an instance belongs to. However, each model's performance varies between the same epoch numbers, with some models showing more substantial change. The Alexnet and Lenet accuracies remain high and show constant decrease in data loss at all epochs. While VGG16 demonstrate less performance, it takes more epochs to get to the same accuracy levels.



(a)	(b)	(c)

FIGURE. 5.3.3. ACCURACY OF EACH MODEL VS EPOCH



(a) (b) (c)

FIGURE. 5.3.4. LOSS OF EACH MODEL VS EPOCH

CHAPTER 6
CONCLUSION
AND FUTURE WORK

CHAPTER 6

CONCLUSION AND FUTURE WORK

6.1. CONCLUSION

The advanced deep learning models, including CNNs, VGG16, AlexNet, and LeNet exhibits considerable potential for sports shot classification using image datasets. Multiple experiments and evaluations show that these models enable the extraction of distinctive features, thereby improving classification accuracy. CNNs serve as the foundation for hierarchical feature detection, contributing to the ability to differentiate between various types of sports shots. The The VGG16, AlexNet, and LeNet designs use deep convolutional networks tuned for feature extraction to improve classification. VGG16 facilitates the extraction of intricate shot-specific features through its deep architecture and small convolutional filters. AlexNet enhances shot-type detection by introducing effective feature learning with deeper layers and ReLU activations. LeNet, with its simple yet effective structure, enables fundamental pattern recognition, contributing to accurate shot classification by capturing essential spatial hierarchies. Overall, these deep learning models can automate and enhance sports shot classification, contributing to advancements in sports analytics, player performance assessment, and broadcasting technologies.

6.2. FUTURE ENHANCEMENT

Future work in the domain of sport shot classification using deep learning approaches should aim to refine and expand the system's capabilities. Fine-grained classification, delving into subtleties within shot categories, could offer a more nuanced understanding of player techniques and styles. Exploring multimodal fusion, incorporating audio or player biometrics, may contribute to a richer contextual analysis. Investigating transfer learning and domain adaptation methods could facilitate quicker adaptation to new sports, while real-time feedback mechanisms and continuous learning approaches would ensure the system remains dynamic and responsive. Additionally, the implementation of edge computing could reduce latency, making the model more versatile. Ethical considerations, such as privacy protection through techniques like federated learning, should be addressed. Collaborating with sports analytics platforms and expanding the dataset to include shots from emerging sports would further enhance the system's utility, fostering its continued relevance and applicability in diverse sports scenarios.

APPENDICES

APPENDICES

A.1. SDG GOALS

A Deep Learning-Based Sports Shot Classification System can align with several United Nations Sustainable Development Goals (SDGs) by contributing to technological advancement, health promotion, and industry innovation. Here's how it maps to specific SDG goals:

1. SDG 3: Good Health and Well-being

Impact:

Enhances athlete performance analysis by providing accurate shot classification and movement analysis. Assists in injury prevention by identifying incorrect techniques or risky moves. Contributes to sports science research, promoting physical fitness and healthy practices.

2. SDG 4: Quality Education

Impact:

Supports sports education and training programs by providing AI-driven insights into player performance. Offers personalized coaching through automated shot assessment. Helps students and trainees understand game strategies through data visualization and classification models.

3. SDG 9: Industry, Innovation, and Infrastructure

Impact:

Promotes technological innovation in sports analytics through deep learning models. Enhances smart stadiums and sports broadcasting by automating shot recognition and event classification. Encourages research and development in AI-based sports systems, fostering innovation.

4. SDG 10: Reduced Inequalities

Impact:

Ensures equal access to performance analysis for athletes of all levels, including grassroots and amateur players. Promotes inclusivity by removing bias in sports performance evaluation through objective AI-driven analysis.

5. SDG 11: Sustainable Cities and Communities

Impact:

Improves the efficiency of sports event management by automating shot classification in real time. Enhances fan engagement and experience through AI-powered sports insights in broadcasts and live events.

6. SDG 17: Partnerships for the Goals

Impact:

Promotes collaboration between sports organizations, tech companies, and research institutions. Supports data-sharing agreements for better sports analysis and technological advancements.

A.2. SOURCE CODE

1. Manual Network

```
# Import the necessary libraries.

import numpy as np

import matplotlib.pyplot as plt

import os

# Import the necessary libraries.

import tensorflow as tf
```

```
import glob
from PIL import Image
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Conv2D, Dense, MaxPooling2D, Flatten, Convolution2D, Dropout, BatchNormalization
# Ignoring the warings
import warnings
warnings.filterwarnings("ignore")
    all_item_dir = os.listdir(item_dir)
    item_files = [os.path.join(item_dir, file) for file in all_item_dir][:n]
    plt.figure(figsize=(35, 10))
    for idx, img_path in enumerate(item_files):
        plt.subplot(2, n, idx+1)
        img = plt.imread(img_path)
        plt.imshow(img, cmap='gray')
        plt.axis('off')
    plt.tight_layout()
def Images_details_Print_data(data, path):
    print(" ====== Images in: ", path)
    for k, v in data.items():
        print("%s:\t%s" % (k, v))
```

```
def Images_details(path):
    files = [f for f in glob.glob(path + "**/*.*", recursive=True)]
    data = {}
    data['images_count'] = len(files)
    data['min_width'] = 10**100  # No image will be bigger than that
    data['max_width'] = 0
    data['min_height'] = 10**100  # No image will be bigger than that
    data['max_height'] = 0
    for f in files:
        im = Image.open(f)
        width, height = im.size
        data['min_width'] = min(width, data['min_width'])
        data['max_width'] = max(width, data['max_height'])
        data['min_height'] = min(height, data['min_height'])
        data['max_height'] = max(height, data['max_height'])
    Images_details_Print_data(data, path)
plot_images(dir_name_train_Toe)
### Data augmentation
train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
# having train and test images
```

```
training_set=train_datagen.flow_from_directory('datasets1/train', batch_size=32, class_mode='categorical', target_size=(255,255))

testing_set=test_datagen.flow_from_directory('datasets1/test', batch_size=32, class_mode='categorical', target_size=(255,255))

## Creating cnn layer

# initialized the model

model=Sequential()

# Adding first convolution layer

model.add(Convolution2D(32,(3,3),input_shape=(255,255,3), padding='same', activation='relu'))

# Adding first Maxpool layer

model.add(MaxPooling2D(pool_size=(2,2)))

# Adding Flatten layer

model.add(Flatten())

# Adding input layer

model.add(Dense(units=64, activation='relu', ))

# Adding dropout layer

model.add(Dropout(0.4))

# Adding output layer

model.add(Dense(units=12, activation='softmax'))

model.summary()

# Compilin the model
```

```
model.compile(optimizer='adam',                              loss='categorical_crossentropy',
metrics=['accuracy'])

# Fitting the model

history= model.fit(training_set,

        steps_per_epoch= training_set.samples//32,

        validation_data=testing_set,

        validation_steps= testing_set.samples//32,

        epochs=20)

def plot():

    # Plot training & validation accuracy values

    plt.plot(history.history['accuracy'])

    plt.plot(history.history['val_accuracy'])

    plt.title('Model accuracy')

    plt.ylabel('Accuracy')

    plt.xlabel('Epoch')

    plt.legend(['Train', 'Test'], loc='upper left')

    plt.show()

    # Plot training & validation loss values

    plt.xlabel('Epoch')

    plt.legend(['Train', 'Test'], loc='upper left')

    plt.show()

plot()
```

2. VGG:

```
import warnings
warnings.filterwarnings('ignore')
import os
import glob
import numpy as np
import pandas as pd
from PIL import Image
import numpy as np
import matplotlib.pyplot as plt
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,
horizontal_flip=True)
training_set=train_datagen.flow_from_directory('datasets/train',target_size=(224,224),
batch_size=32,class_mode='categorical')
test_datagen=ImageDataGenerator(rescale=1./255)
test_set=test_datagen.flow_from_directory('datasets/test',target_size=(224,224),batch_
size=32,class_mode='categorical')
model = Sequential()
model.add(Conv2D(input_shape=(224,224,3), filters=64, kernel_size=(3,3),
padding="same", activation="relu"))
model.add(Conv2D(filters=64, kernel_size=(3,3), padding="same", activation="relu"))
model.add(MaxPool2D(pool_size=(2,2), strides=(2,2)))
```

```
model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))

model.add(Conv2D(filters=128, kernel_size=(3,3), padding="same", activation="relu"))

model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Conv2D(filters=256, kernel_size=(3,3), padding="same", activation="relu"))

model.add(MaxPool2D(pool_size=(2,2),strides=(2,2)))

model.add(Flatten())

model.add(Dense(256, activation='relu'))

model.add(Dense(128, activation='relu'))

model.add(Dense(20, activation='softmax'))

model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])

model.summary()

es = EarlyStopping(monitor = 'accuracy', patience = 10)

mc = ModelCheckpoint('vggmodel.h5', monitor = 'accuracy', verbose=1, save_best_only = True)

epochs = 5

batch_size = 32

#### Fitting the model

history = model.fit(

    training_set, steps_per_epoch=training_set.samples // batch_size,

    epochs=epochs,
```

```
        validation_data=test_set,validation_steps=test_set.samples // batch_size,
        callbacks=[es, mc])
plt.plot(history.history['accuracy'])
plt.plot(history.history['val_accuracy'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.show()
```

3. LeNet Architecture

```
# Import the warnings
import warnings
warnings.filterwarnings("ignore")
# Import the neccesary Packages.
```

```
import os

import glob

import numpy as np

import pandas as pd

from PIL import Image

import matplotlib.pyplot as plt

train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,
horizontal_flip=True)

training_set=train_datagen.flow_from_directory('datasets2/train',target_size=(224,224)
,batch_size=32,class_mode='categorical')

test_datagen=ImageDataGenerator(rescale=1./255)

test_set=test_datagen.flow_from_directory('datasets2/test',target_size=(224,224),batch
_size=32,class_mode='categorical')

Classifier=Sequential()

Classifier.add(Convolution2D(32,3,3,input_shape=(224,224,3),activation='relu'))

Classifier.add(MaxPooling2D(pool_size=(2,2)))

Classifier.add(Convolution2D(128,3,3,activation='relu'))

Classifier.add(MaxPooling2D(pool_size=(2,2)))

Classifier.add(Flatten())

Classifier.add(Dense(256, activation='relu'))

Classifier.add(Dense(20, activation='softmax'))

Classifier.compile(optimizer='rmsprop',loss='categorical_crossentropy',metrics=['accur
acy'])
```

```
Classifier.summary()
# Set the file path
model_path = "LeNet1.h5"
callbacks = [
   ModelCheckpoint(model_path, monitor='accuracy', verbose=1, save_best_only=True)
]
epochs = 100
batch_size = 32
#### Fitting the model
history = Classifier.fit(
         training_set, steps_per_epoch=training_set.samples // batch_size,
         epochs=epochs,
         validation_data=test_set,validation_steps=test_set.samples // batch_size,
         callbacks=callbacks)
plt.figure(figsize=(20, 8))
plt.plot(history.history['accuracy'])
for i in range(epochs):
   if i%5 == 0:
plt.annotate(np.round(history.history['accuracy'][i]*100,2),xy=(i,history.history['accuracy'][i]))
plt.title('Model accuracy')
plt.ylabel('Accuracy')
```

```
plt.xlabel('Epoch')

plt.show()

plt.figure(figsize=(20, 8))

plt.plot(history.history['loss'])

for i in range(epochs):

    if i%5 == 0:

  plt.annotate(np.round(history.history['loss'][i]*100,2),xy=(i,history.history['loss'][i]

plt.title('Model Loss')

plt.ylabel('Loss')

plt.xlabel('Epoch')

plt.show()
```

4. Alexnet Architecture

```
# Import the neccessary Libraries.

import numpy as np

import matplotlib.pyplot as plt

# Import the neccesary Libraries.

import warnings

warnings.filterwarnings("ignore")

# Import the neccesary packages.

import glob

import tensorflow as tf
```

```
from tensorflow.keras.layers import BatchNormalization
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.models import Sequential
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.callbacks import ModelCheckpoint
# Data Augumentation.
train_data_gen = ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range=0.2, horizontal_flip=True)
test_data_gen = ImageDataGenerator(rescale= 1./255)
# Splitting train and test
training_set=train_data_gen.flow_from_directory('datasets1/train',target_size=(224,224),batch_size=32,class_mode='categorical')
test_set=test_data_gen.flow_from_directory('datasets1/test',target_size=(224,224),batch_size=32,class_mode='categorical')
# Create a sequential model
model = Sequential()
# 1st Convolutional Layer
model.add(Conv2D(filters=96, input_shape=(224,224,3), kernel_size=(11,11),\
 strides=(4,4), padding='valid'))
model.add(Activation('relu'))
# Pooling
model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))
# Batch Normalisation before passing it to the next layer
```

```
model.add(BatchNormalization())

# 2nd Convolutional Layer

model.add(Conv2D(filters=256, kernel_size=(11,11), strides=(1,1), padding='valid'))

model.add(Activation('relu'))

# Pooling

model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))

# Batch Normalisation

model.add(BatchNormalization())

# 3rd Convolutional Layer

model.add(Conv2D(filters=384, kernel_size=(3,3), strides=(1,1), padding='valid'))

model.add(Activation('relu'))

model.add(Activation('relu'))

# Batch Normalisation

model.add(BatchNormalization())

# 5th Convolutional Layer

model.add(Conv2D(filters=256, kernel_size=(3,3), strides=(1,1), padding='valid'))

model.add(Activation('relu'))

# Pooling

model.add(MaxPooling2D(pool_size=(2,2), strides=(2,2), padding='valid'))

# Batch Normalisation

model.add(BatchNormalization())

# Passing it to a dense layer
```

```
model.add(Flatten())

# 1st Dense Layer

model.add(Dense(4096, input_shape=(224*224*3,)))

model.add(Activation('relu'))

# Add Dropout to prevent overfitting

model.add(Dropout(0.4))

# Batch Normalisation

model.add(BatchNormalization())

# 2nd Dense Layer

model.add(Dense(4096))

model.add(Activation('relu'))

# Add Dropout

model.add(Dropout(0.4))

# Batch Normalisation

model.add(BatchNormalization())

# 3rd Dense Layer

model.add(Dense(1000))

model.add(Activation('relu'))

# Add Dropout

model.add(Dropout(0.4))

# Batch Normalisation

model.add(BatchNormalization())
```

```
# Output Layer
model.add(Dense(12))
model.add(Activation('softmax'))
# Compile the model
model.compile(loss = 'categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
model.summary()
epochs = 40
batch_size = 32
def plot():
    # Plot training & validation loss values
    plt.plot(History.history['loss'])
    plt.plot(History.history['val_loss'])
    plt.title('Model loss')
    plt.ylabel('Loss')
    plt.xlabel('Epoch')
    plt.legend(['Train', 'Test'], loc='upper left')
    plt.show()
plot()
```

A.3. SCREENSHOTS

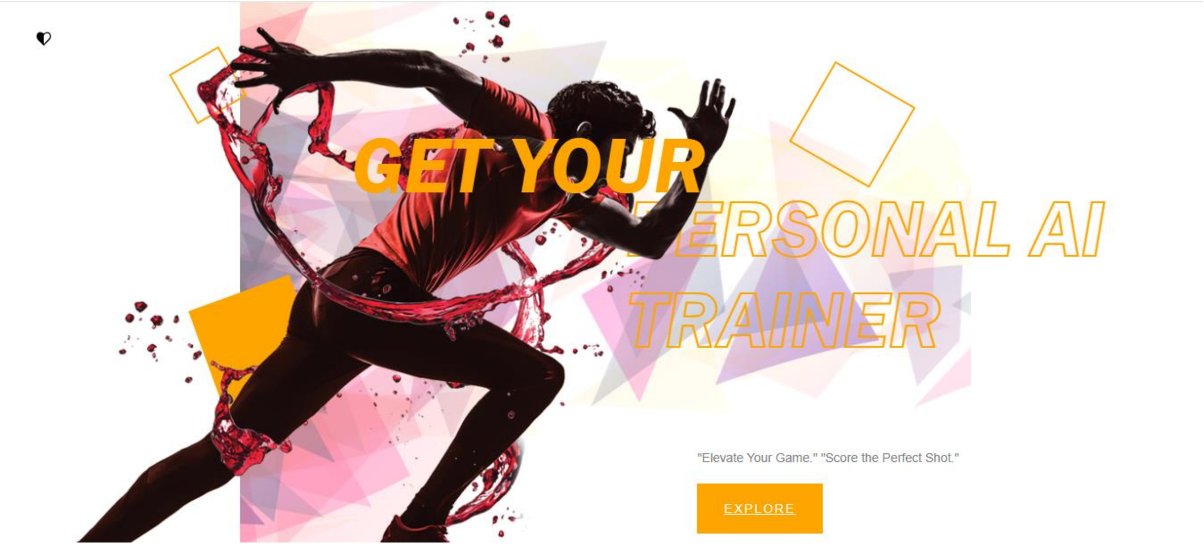


FIG A.3.1 HOME PAGE

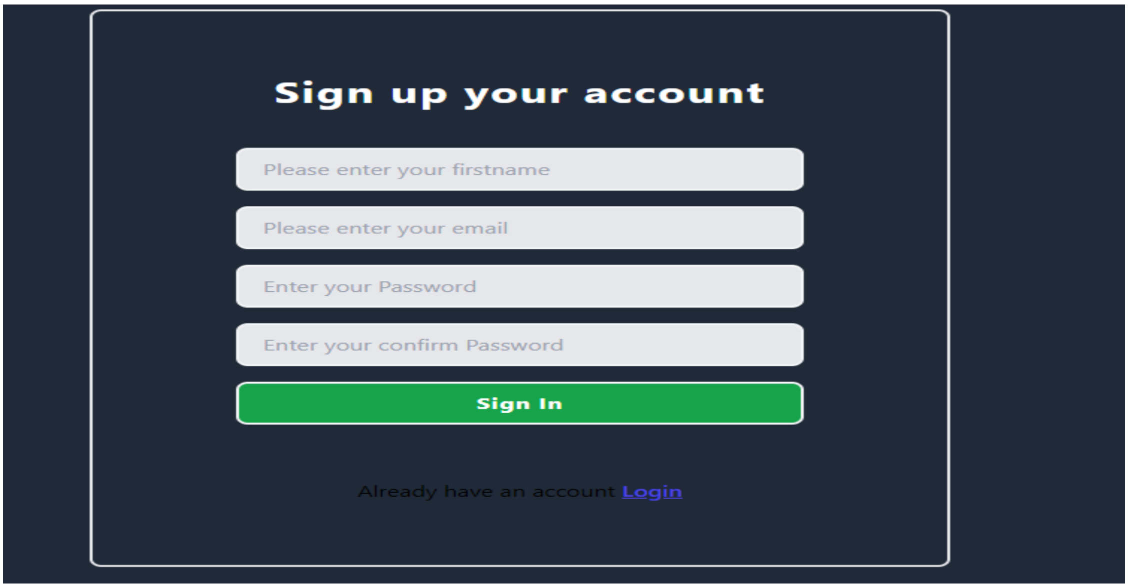


FIG A.3.2 SIGN-IN PAGE

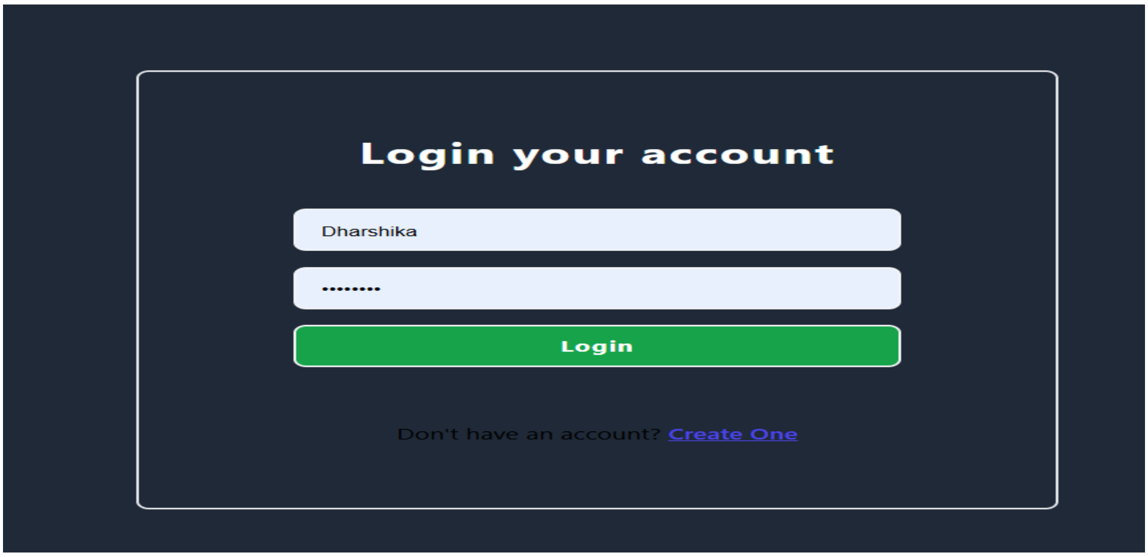


FIG A.3.3 LOGIN PAGE

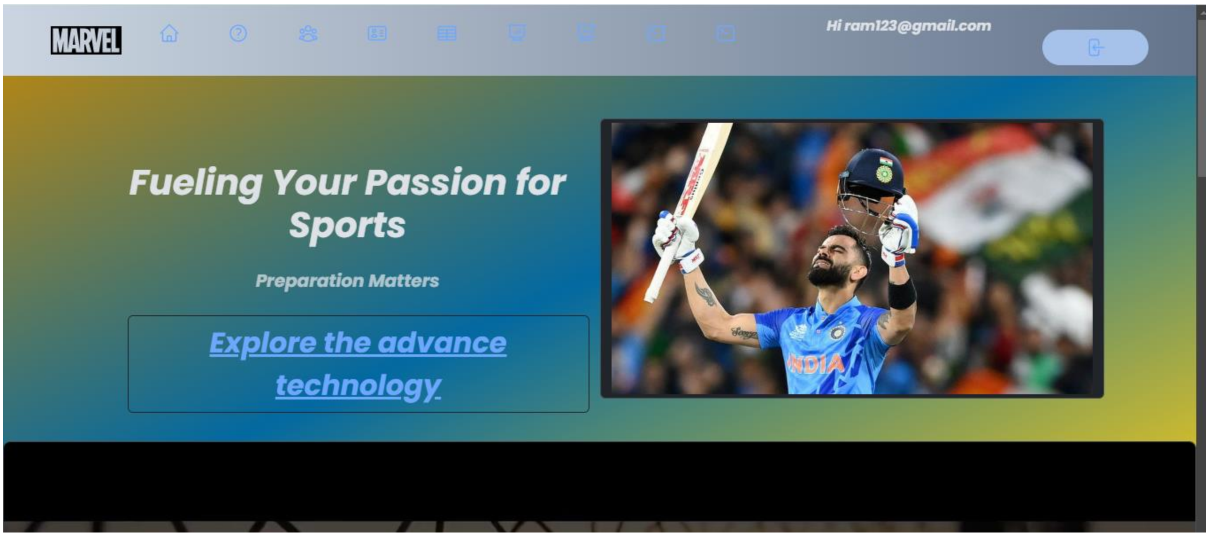


FIG A.3.4 HOME PAGE AFTER LOGIN

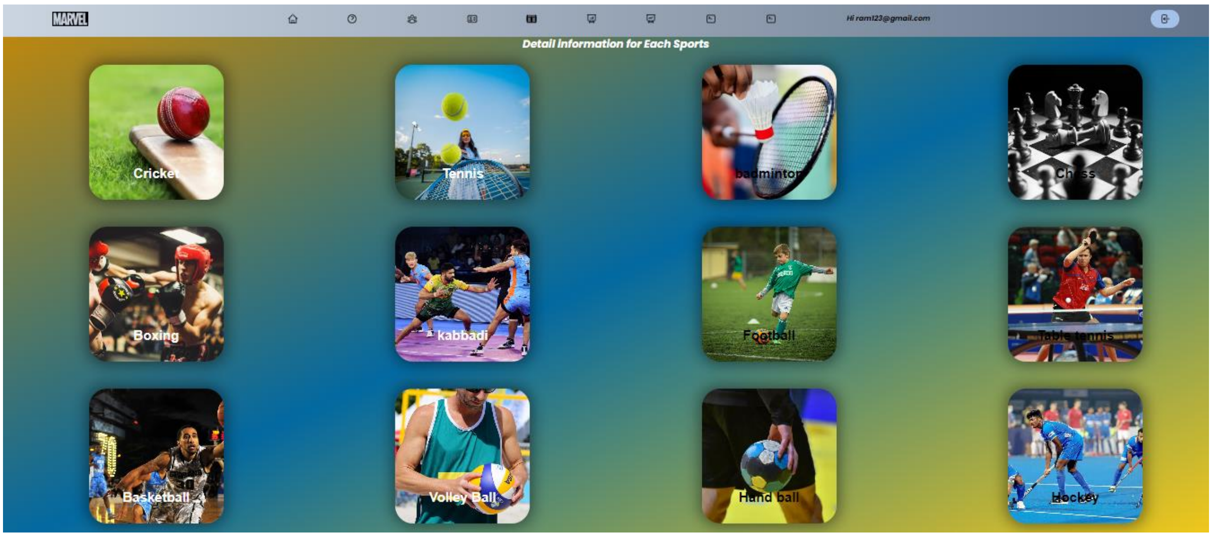


FIG A.3.5 DETAIL INFO OF EACH SPORTS

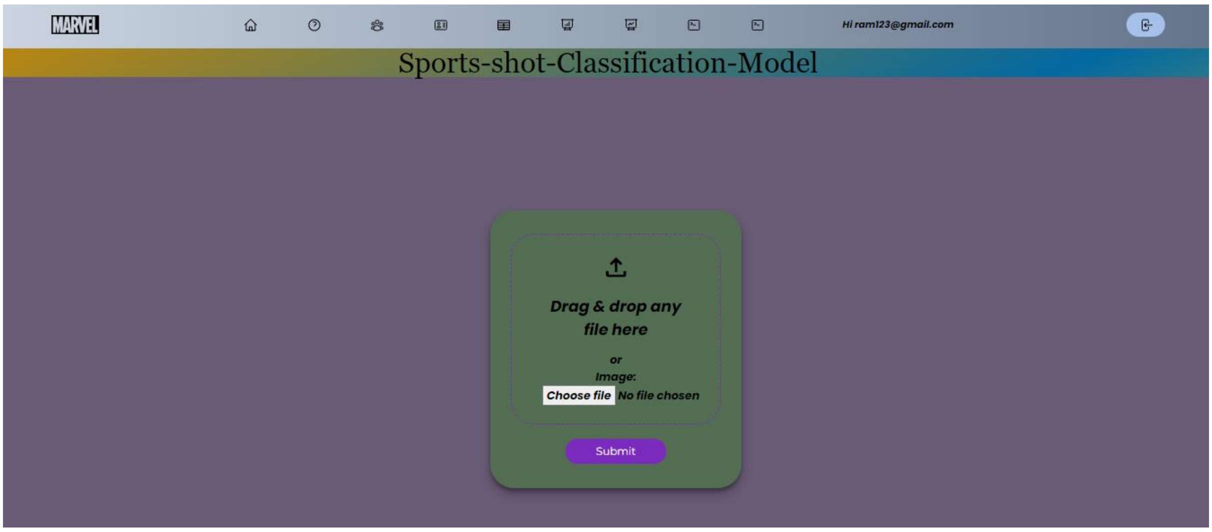


FIG A.3.6 MODULE 1(INSERT IMAGE FROM DATASET 1)

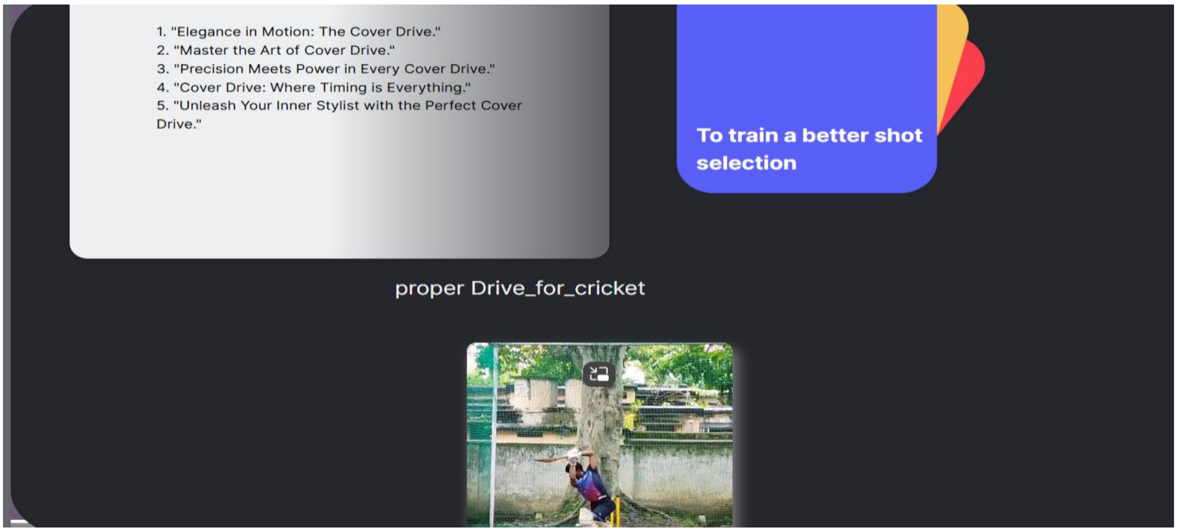


FIG A.3.7 OUTPUT PAGE OF MODULE 1

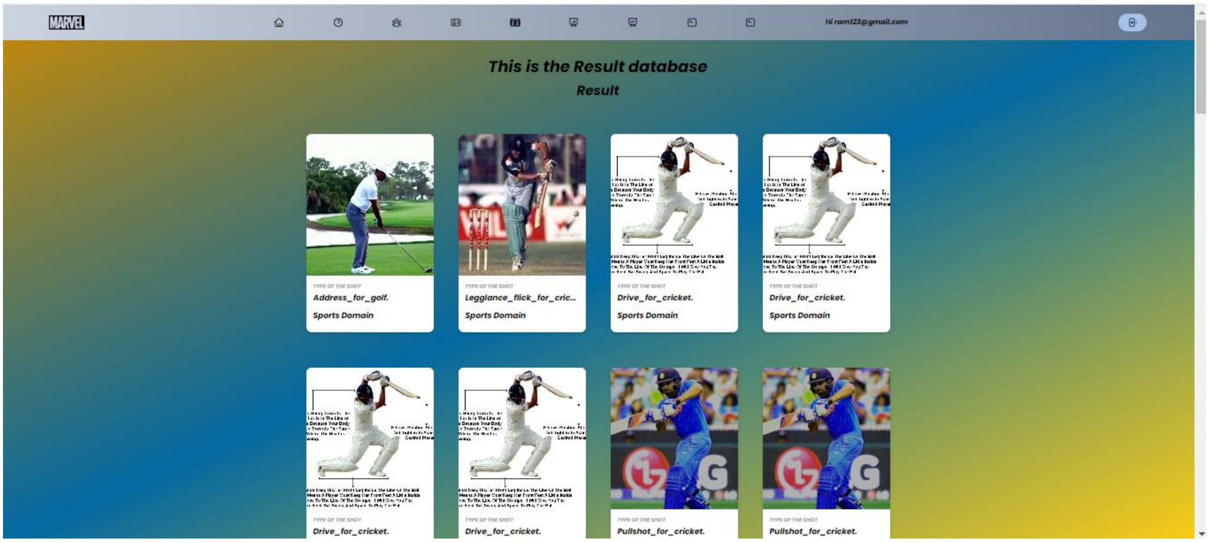


FIG A.3.8 RESULT DTATBASE OF MODULE 1

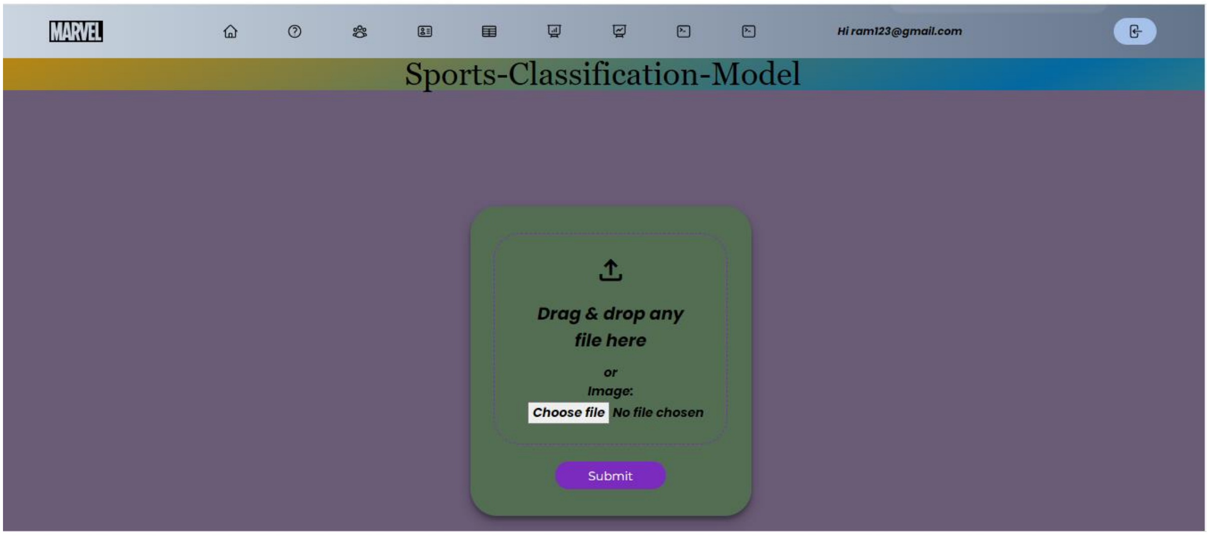


FIG A.3.9 MODULE 2 (INSERT IMAGE FROM DATASET 2)

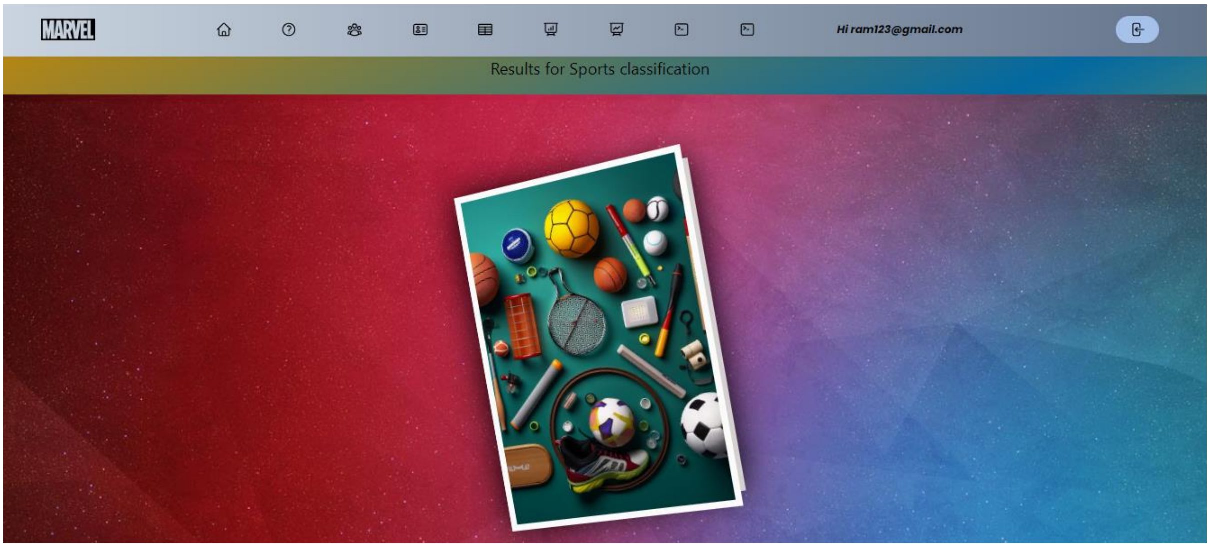


FIG A.3.10 OUTPUT PAGE OF MODULE 2

FIG A.3.11 DESCRIPTION PAGE

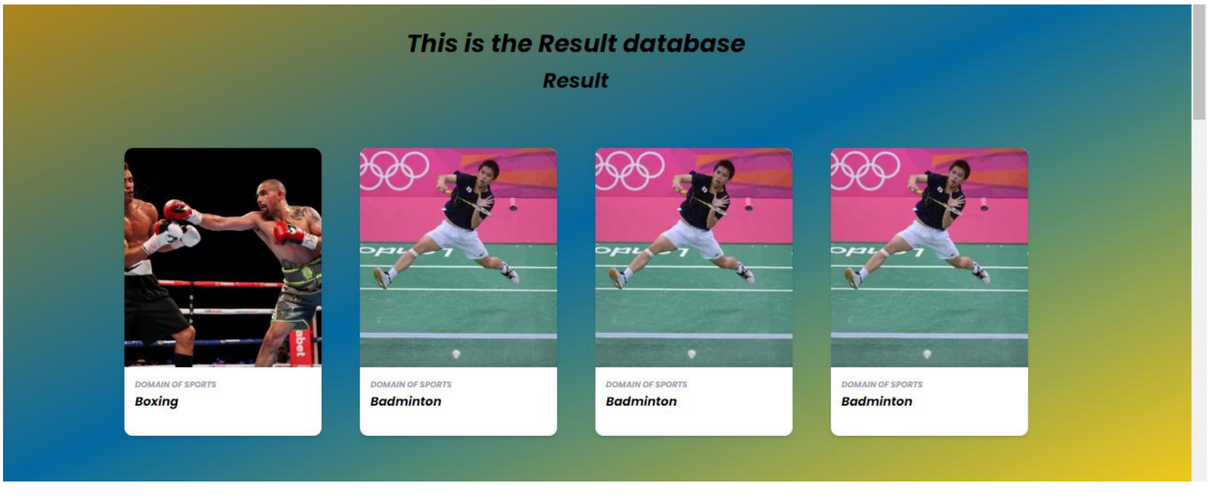


FIG A.3.12 RESULT DTATBASE OF MODULE 2

A.4. PLAGIARISM REPORT

Deep Learning-Based Sports Shot Classification System

Dr.Vinmathi M S[1]
Professor
Department of Computer Science and Engineering
Panimalar Engineering College
vinmathis@gmail.com

Hari Priya R[3]
Department of Computer Science and Engineering.
Panimalar Engineering College
priyaranganathan2003@gmail.com

Dharshika S[4]
Department of Computer Science and Engineering.
Panimalar Engineering College
bsdi1246789@gmail.com

Logeshwari S[5]
Department of Computer Science and Engineering.
Panimalar Engineering College
logeshwarisrinivasan5@gmail.com

Abstract- This study presents an innovative deep learning architecture for sports shots classification, employing Convolutional Neural Networks (CNNs) developed using TensorFlow. Accurate shot classification is crucial for enhancing sports analytics, coaching methodologies, and broadcasting, as it yields critical insights into gameplay dynamics and athlete performance. Conventional techniques often rely on manual labeling or basic rule based system, which is insufficiently adaptable to complexities of different sports. We explore a range of CNN architectures specifically optimized for shot classification, enhancing model efficacy through strategies such as data augmentation, advanced deep learning techniques, and meticulous hyperparameter tuning. The use of TensorFlow ensures robust scalability and efficiency in both the training and deployment phases of the model. Our proposed framework shows significant promise for transforming sports analysis, broadcasting, and coaching practices, thereby enabling automated and sophisticated shot classification across diverse sporting contexts.

Keywords— ***Deep learning, CNN, AlexNet, LeNet, VGG16, Sports Shot images***

I. INTRODUCTION

Advances in data collecting and analysis technology have led to a considerable evolution in sports analytics in the past few years. Accurate classification of sports shots is crucial for enhancing sports analytics, coaching methodologies, and broadcasting, as this offers vital information about gameplay dynamics and athlete performance, essential for making informed decisions. Traditional techniques for shot classification often rely on manual labeling or basic rule-based systems. While useful to some extent, these methods have significant limitations, including being labor-intensive, time-consuming, and inflexible in handling the complexities and variations inherent in different sports. Recent advancements in Convolutional Neural Networks(CNNs), a kind of deep learning, have demonstrated significant potential in tackling these issues. CNNs immediately learn and extract relevant features from data, achieving high accuracy in classification tasks and adapting to the complexities of various sports. This study presents an innovative deep learning framework for the classification of sports shots using CNNs developed with TensorFlow. Our approach includes exploring a range of optimized CNN architectures, employing data augmentation techniques, incorporating advanced deep learning methods such as dropout, batch normalization, transfer learning, and meticulously tuning hyperparameters to enhance model efficacy. By leveraging TensorFlow, we ensure robust scalability and efficiency during the model's training and deployment stages. Our proposed framework represents a noteworthy development in sports analytics, providing an opportunity to transform sports analysis, broadcasting, and coaching practices by automating and sophisticating shot classification. This, in turn, seeks to offer deeper information about athlete performance as well gameplay strategies, ultimately contributing to the evolution of sports analytics.

II. LITERATURE REVIEW

Several approacheshave been investigated in the area of sports shot recognition, indicating the development of methods in this field.The results were compiled and interpreted using Convolutional Neural Network (CNN)-based lightweight technique for re-play identification in cricket videos [1]. Linear SVM camera models with a Random Forest models were used for temporal action localisation in Cricket Stroke Extraction, another noteworthy contribution [2].However, these approaches lacked extensive consideration of diverse stroke styles, limiting their applicability across broader datasets.

In Outcome Classification in Cricket Using Deep Learning [3], a ball by ball outcome prediction method achieved a notable accuracy of 70%, although it was hindered by the absence of a standardized dataset. Such challenges have prompted researchers to explore attention mechanisms and curated datasets to overcome variations in cricketing conditions.In this field, deep learning-based methods have shown a great deal of promise. The study "Deep CNN-based

Data-driven Identification of Batting Shots in Cricket" [4] utilized a combination of RNN, 3D CNN, and 2D CNN architectures to achieve high level of accuracy. However, it faced restrictions in robustness under diverse conditions and relied on small, imperfect datasets. Likewise, "CricShotClassify" [5] demonstrated a CNN-GRU hybrid architecture with a remarkable 93% accuracy rate. This method addressed misclassifications caused by extraneous frames and varying angles but could be extended further through enhanced dataset creation and advanced network designs. To show the effectiveness of hybrid deep learning models over sports action recognition, Ahmad et al. [6] suggested an optimised CNN-LSTM network to categorise five distinct cricket shots—bowled, defence, cover, pull, and reverse shots from films. Foysal et al. [7] shown the usefulness of convolutional networks in static frame-based action identification by using a CNN-based method to categorise various cricket shots from photos. Ahmed et al. [8] employed a CNN model to classify four distinct cricket batting strokes from videos, emphasizing the role of deep learning in automating cricket action recognition. Sen et al. [9] demonstrated the benefits of integrating convolutional and recurrent networks for sequential data processing by introducing a CNN-GRU architecture to identify batting shots from cricket films.

Gupta et al. [10] developed a method for cricket stroke classification by leveraging spatial and motion sequences, reinforcing the importance of spatiotemporal information in shot identification.In order to classify cricket strokes, Azhar et al. [11] proposed a transformer model derived from DenseNet, proving the effectiveness of transformer architectures in sports analytics. Devanandan et al. [12] utilized a Random Forest model for identifying cricket shots from images, presenting a machine learning approach as an alternative to deep learning for sports classification tasks. Thomas et al. [13] and Wang et al. [14] present comprehensive surveys on both contemporary and past research in the field. While datasets such as Sports-1M and UCF Sports [15] are among the biggest one for sports analysis, they lack sufficient data to effectively train models for a specific sport, as they do not encompass all types of events within any single sport.A motion-estimation method was presented in [16] for the classification of cricket shots. In contrast to deep learning models, this approach's accuracy was lower yet it identified eight angle-based classes to identify various shots. The off drive and hook shot had the highest reported accuracy rates, with 63.57% and 53.32%, respectively. In [17], cricket shots were categorised using wearable technology. The objective was to develop a method for quality assessment.The objective was to develop a system for evaluating quality that took into account factors like shot direction and foot position. Five layers of action classification were achieved by hierarchical representation using (k-NN) k-nearest neighbours, (SVM) support vector machines, and decision trees (DT). The best-performing classifiers' class-weighted F1 score was 88.30%. Nevertheless, this study did not evaluate actual match situations; it simply used films of training sessions.

For both left- handed and right-handed batsmen, Semwal et al. [18] created an example to identify batting strokes such the straight drive, lofted on drive, and cover drive. They achieved 83.098% accuracy for right-handed shots and 65.186% accuracy for left-handed shots using a pretrained CNN and a multiclass SVM. However, some shots, including the left cover drive, left straight drive, and right straight drive, were misclassified due to a lack of data.A method for categorising cricket shots into the run, four, six, and out occurrences—was presented in [19]. An SVM was used to classify the individual shots that were extracted from the movie.The accuracy of the suggested model was 87.8%, while it performed worse for six-hit occurrences. Using an AlexNet CNN-based technique, Rafiq et al. [20] presented based on transfer learning system for sports video summarisation that obtained 99.26% accuracy on a smaller dataset that contained only cricket sequences.

III. METHODOLOGY

The sports shot images were thoroughly collected for this research to create an extensive dataset. A total of 4000 image data of the sport short were collected which were classified by the type of sport short. A total of 4,000 sports shot images were collected and categorized into 12 classifications based on different stages and types of shots. This classification is crucial because it provides a diverse set of deep learning model's training data, which helps the capacity of the model to recognize and categorize sports shots in images. In particular, there are 400 images for testing and each of the eight classifications-address, impact, finish, mid-downswing, mid-backswing, and mid-followthrough, toe-up, and top has 599 images for training. The dataset, which is part of the leg glance flick and sweep class, includes 600 training pictures and 299 testing images. For the drive and pull shot classes, there will be 299 testing images and 599 training images. Therefore, having such image datasets would enable the model to examine and predict the image in the future.

In the research study, several DL models were used to predict and analyze the Sport shot image data of the various sport for its classification. Figure 1 shows the various DL models used and the methodology of the research. These include the CNNs, VGG16, AlexNet and LeNet. These models have shown better results in the sports domain on the image dataset. The CNN model captured the hierarchical and spatial feature in the image as the foundational architecture. The AlexNet and LeNet model had deeper architecture to do better feature extraction and achieve better accuracy. VGG16 had a sophisticated architecture network with efficient computation.

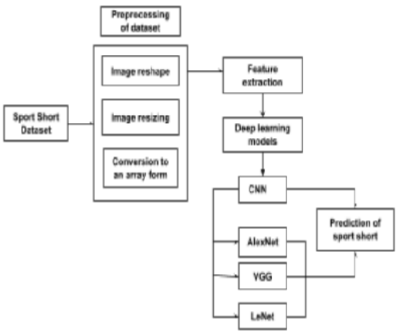


Fig. 1. Methodology of the research

Finally, AlexNet prevents the vanishing gradient exception in bigger neural networks so that we can go deeper without the loss of their performance. The implementation of the model involved various stages such as data preprocessing to make data more normalized and augmentation to maximize the quality and variance in the image dataset. These images were then trained and validated, where the high classes were learned manually by the model. Validation involved examining the model to ensure generalization of the data unseen during training. Finally, after evaluating the model's performance based on F1-Score, accuracy, and recall. Predicting the sport shot classes and improving player performance are the goals of this study. Since players' performance raises their scores, it is necessary to improve results.

IV. DEEP LEARNING MODELS

Sports shot categorization is a difficult task due to the dynamic movements, varying angles, and environmental conditions that affect shot recognition. Models for deep learning, particularly about Convolutional Neural Networks (CNNs) have proved notable improvements in recognizing different shot types across sports. This study employs AlexNet, VGG16, and LeNet to classify sports shots, comparing their accuracy to decide which model is most suited for this task.

In sports shot classification, AlexNet excels at recognizing body posture, ball trajectory, and shot execution. AlexNet is a CNN model designed for high-accuracy image classification. Each of its five convolutional layers is succeeded by max pooling and ReLU activation layers, which enable the extraction of spatial and motion features.

The Visual Geometry Group at the University of Oxford developed the sophisticated CNN architecture known as VGG16. It accepts 224x224-pixel images as input, maintaining consistency for the ImageNet competition. The model uses 3x3 convolutional filters to capture spatial movements and ReLU activation to speed up training. Three fully connected layers come after the VGG16 convolutional layers, with 4096 channels in the first two and the third layer having 1000 channels for classification. Local response normalization is typically not used in VGG16 to avoid increased memory usage and training time.

LeNet is one of the simplest convolutional neural network introduced by Yann LeCun in 1989, commonly referred as LeNet-5. LeNet, originally designed for digit recognition, serves as a lightweight baseline model for sports shot classification. Its two layers of convolution are then subsampling (pooling) layers enabling efficient extraction of features, making it useful when dataset size is limited or computational resources are constrained.

Table 1 represents in-depth analysis of the three model architectures-AlexNet, VGG16, and LeNet which were employed in this study. In order to prevent overfitting, AlexNet uses eight layers with 11x11, 5x5, and 3x3 filters. Max-pooling is used after the first two convolutional layers, dropout is used in fully connected layers and ReLU activation is used in all levels except output. With 16 layers and uniform 3x3 filters, VGG16 uses fully connected layers for classification, max-pooling after convolutional layers, and ReLU activation in hidden layers. LeNet, which has five layers, uses 5x5 and 3x3 filters. Tanh activation was used at first, but ReLU was later used for superior performance. Subsampling is used to minimize spatial dimensions and effectively extract features

Model	Layers	Filter Size	KeyFeatures
AlexNet	8	11×11, 5×5, 3×3	Max-pooling, Dropout, ReLU
VGG16	16	3×3	ReLU, Max-pooling, Fully Connected
LeNet	5	5×5, 3×3	Tanh, Subsampling

Table 1. Comparison of model architecture.

By leveraging VGG16 domain-specific adaptability, AlexNet's deep feature extraction, and LeNet's efficiency, this study provides a comprehensive approach to sports shot classification. Figure 2 represents that the highest-performing model (AlexNet) is selected for final implementation, ensuring accurate recognition of diverse shot types, aiding in performance analysis, coaching, and automated sports analytics.

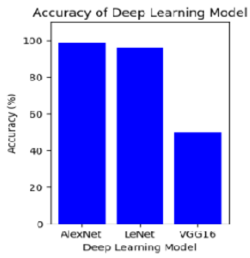


Fig. 2. Accuracy of each model

V. PREPROCESSING OF DATASET

Data preprocessing is essential to ensuring the success of deep learning models for sports shot classification. Since raw images often contain variations in quality, resolution, and lighting conditions, careful preprocessing strengthens the model's capacity to extract meaningful features, improving classification accuracy and overall performance.

The first step involves collecting and cleaning the datasets. Sports action images are gathered from multiple sources, including publicly available datasets and manually labeled images. These images capture various shot types performed by athletes in different sports. During the cleaning process, low-quality, blurry, or corrupted images are removed to prevent inaccuracies in feature extraction. Ensuring high-quality data is crucial, as noisy or distorted images can negatively impact model training and lead to misclassifications. Once the dataset is cleaned, normalization is applied to bring all images to a standard intensity range.

Pixel values are scaled between 0 and 1 to maintain consistency and improve training efficiency.

Without normalization, variations in pixel intensities may cause instability during training, leading to poor generalization on unseen data. This step also helps the deep learning model process images more effectively by ensuring that features extracted from different images remain comparable. To maintain uniformity in input dimensions, all images are resized to a fixed resolution. Since deep learning models operate with predefined input sizes, resizing ensures compatibility without distorting image features. Maintaining aspect ratios where possible prevents loss of important structural details in the images. By standardizing input dimensions, the model can efficiently learn distinguishing characteristics across different shot types. Given the limited availability of sports action datasets, data augmentation is applied to artificially expand the training set. Transformations such as rotation, flipping, scaling, and brightness adjustments introduce variability while preserving essential features. Augmentation reduces overfitting, allowing the model to generalize better to new images. Since athletes perform similar shots under different lighting conditions, angles, and orientations, augmentation ensures that the model learns from diverse representations rather than memorizing specific examples. To further refine image quality, noise reduction techniques are employed. Sports images, especially those extracted from live-action videos, often contain motion blur and background clutter. Applying Gaussian blurring and median filtering helps remove unwanted artifacts while retaining essential action details. Enhancing contrast ensures that the model focuses on the player's movement and shot mechanics rather than distractions in the background. This preprocessing step improves the clarity of extracted features, leading to more precise classification results.

After preprocessing, a stratified sample technique is used to divide the dataset into training, validation, and test sets. This ensures that each shot type is proportionally represented across all sets, preventing bias toward majority classes. Images are utilized to identify patterns in the training set, and the validation set aids hyperparameter tuning. In contrast, the test set evaluates how well the finished model performs on unknown data. Proper dataset splitting ensures that the trained model remains robust and performs well across different scenarios. Following preprocessing, multiple deep learning models are trained and evaluated on the processed dataset. The best model for classifying sports shots is identified using performance criteria such as accuracy, precision, recall, and F1 score. The model achieving the highest accuracy on the test set is selected for final deployment. By ensuring a structured preprocessing pipeline, the classification system becomes more reliable, improving its ability to recognize and differentiate between various sports actions.

VI. FEATURE EXTRACTION

This is a crucial step in cricket shot classification, as it enables the deep learning model to identify and differentiate between different shot types based on player movements, bat trajectory, and ball position. This process involves applying convolutional neural networks (CNNs) such as AlexNet, VGG16, and LeNet to extract meaningful patterns from the video frames. These architectures improve the ability of the model to depict both low-level and high-level spatial features, improving classification accuracy.

AlexNet, a convolutional neural network, is widely used for image classification tasks. Five convolutional layers and max-pooling layers make up this system, which gradually reduces spatial dimensions while retaining essential features. The first convolutional layer utilizes large 11×11 filters to capture broad spatial patterns, while deeper layers use smaller 3×3 filters to refine the details. Activation functions such as ReLU speed up training by introducing non-linearity, helping the network to learn complex patterns in cricket shots. Dropout layers randomly deactivate neurons during training in order to minimize overfitting, making sure the model performs adequately in generalizing to unknown video frames. Furthermore, the retrieved characteristics are processed for categorization by fully linked layers after the network, which finally gives the cricket shot a label.

VGG (Visual Geometry Group) networks, specifically VGG16 and VGG19, are designed to improve feature extraction by using very small convolutional filters of size 3×3. Unlike AlexNet, which starts with larger filter sizes, VGG relies on a uniform structure with multiple stacked convolutional layers to extract intricate spatial features. The deeper architecture allows the network to capture finer motion details in cricket shots, making it particularly effective for distinguishing visually similar shots like cover drives and square cuts. By maintaining the same filter size throughout the network and increasing the depth, VGG16 enhances the hierarchical learning of features. Max-pooling layers gradually lower spatial dimensions while maintaining significant shot-related properties, and all hidden layers employ the ReLU activation function. At the end, the fully connected layers divide the retrieved representations into several shot types.

LeNet is one of the earliest convolutional neural network, provides a structured yet simpler approach to feature extraction. For classification, the model comprises three completely linked layers, two average-pooling layers, and convolutional layers. LeNet efficiently captures essential patterns with fewer parameters, making it computationally lightweight compared to AlexNet and VGG16. The use of tanh activation in earlier versions was later replaced with ReLU for improved training efficiency. Despite being a smaller network, LeNet remains useful for recognizing fundamental shot patterns, making it a strong baseline model for comparison in cricket shot classification.

Each of these CNN architectures contributes uniquely to feature extraction in cricket shot classification. AlexNet excels in learning both broad and detailed features, making it suitable for complex datasets with diverse shot variations. VGG16's deep and structured architecture refines spatial feature extraction, making it ideal for recognizing intricate shot mechanics. LeNet, though simpler, offers an efficient and lightweight alternative for applications requiring lower computational resources.

VII. RESULT AND DISCUSSION

The different deep learning models' performance is evaluated using the Sport shot Dataset, which consisted of 4000 Sports shots sourced from [source, e.g., Kaggle's Sport shot dataset, 2024]. This dataset included images from twelve distinct classifications and it is divided into four different classifications based on the shots. To make training and assessing models easier, the dataset was separated into training (60%), validation (20%), and testing (20%) sets. Preprocessing included resizing the images to 224x224 pixels and normalizing them to [mean and standard deviation]. In order to reduce overfitting to enhance the generalization of the model, data augmentation methods include flipping, rotation, and random cropping were applied during training. The models were subsequently evaluated on the test set, with performance metrics summarized to demonstrate the comparative efficacy of the architectures.

After training the models, they were thoroughly evaluated using a test dataset to determine their ability to classify sports shots accurately as shown in figure 2. The AlexNet model demonstrated the highest accuracy, achieving 99.70% for training and 96.15% for validation, making it the most effective model for sports shot classification. Its deep architecture, use of ReLU activation, and dropout regularization contribute to its high performance by efficiently extracting features while preventing overfitting. The LeNet model, known for its simplicity and computational efficiency, achieved 96.47% accuracy. LeNet-5, originally introduced by YannLeCun in 1989, comprises two subsampling layers, two fully linked layers, and three convolutional layers. Despite its architecture's relative narrowness compared to AlexNet, it effectively captures spatial features using convolution and tanh activation while maintaining a lower computational cost. The VGG16 model, although deeper than LeNet, yielded an accuracy of 50%, indicating that its increased depth might not be optimal for sports shot classification in this dataset. While VGG16 is known for using small 3×3 convolutional filters to capture fine-grained spatial details, its complexity might require more fine-tuning or a larger dataset for improved performance. These results highlight the effectiveness of AlexNet in sports shot classification due to its superior feature extraction capabilities and training stability. The structured evaluation of these CNN architectures ensures that the model chosen for deployment can accurately classify various sports shots, enhancing automated analysis in sports analytics.

Model	Precision	Recall	F1 Score
AlexNet	96 %	95 %	95.5 %
LeNet	90 %	89 %	89.5 %
VGG16	52 %	51 %	51.5 %

Table. 2. Metrics scores of each model

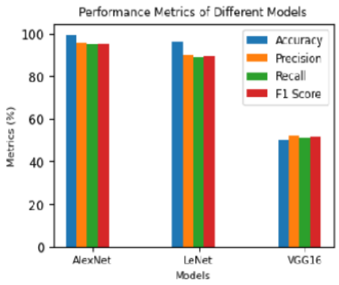


Fig. 3. Performance Metrics of Different Models

Figure 3 represents the value of the performance scores, such as precision, recall, and F1 score for DL models in the study. Precision is defined as the proportion of genuine positive predictions among all of the model's positive predictions. Sensitivity is another name for recall, which is the percentage of true positive models among all positive events in the dataset. However, the F1 score measures the harmonic mean between precision and recall. As such, F1 score is a balanced measure of the models accuracy since it considers both false positives and false negatives. The AlexNet model in Convolutional Neural Network exhibited high performance in all the models,as shown in table 2. Thus, the model has a high degree of capturing and classifying accurate sport shot since the measures for precision and recall were close. The Lenet model is slightly less compare to AlexNet, and one of the last performing models was the VGG16 model, which had slightly low values as compared to the Alexnet, and Lenet. Therefore, the best model was Alexnet with high values across the metric.

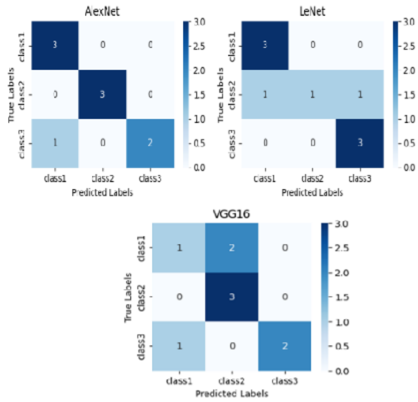


Fig. 4. Confusion matrices of each model

The figure 4 matrixes of confusion displays a thorough, easy-to-read depiction of how each model's predictions performed for various sports shot classifications. The AlexNet displays excellent accuracy, as evidenced by the confusion matrix for this model. Specifically, out of 4000 instances, 599 Address shots were correctly predicted, as were 599 Finish shots, 599 Impact shots, and 599 Mid Backswing shots. In addition, only a few wrong classifications are made, including 12 Finish shots predicted as Address, 11 Impact shot predicted as Finish, and 8 Mid Backswing shot predicted as Impact.

Therefore, this model demonstrates a high level of performance. AlexNet also correctly identifies 600 LegGlance Flick shots, 599 Sweep shots, and 599 Drive shots. Although the error measurements are marginally greater than those aggregated for the AlexNet dataset, this model demonstrates a strong level of accuracy. LeNet represents another high-performing model, as demonstrated by the confusion matrix. LeNet predicts 599 Pull shots, 599 Toe-Up shots, and 599 Top shots. Indeed, the model demonstrates accuracy, with only 10 Toe-Up shots predicted incorrectly. Additionally, VGG16 identifies 599 Mid Follow Through shots, 599 Mid Downswing shots, and 599 Top shots. This model displays respectable levels of accuracy; while the accuracy measurable is lower, the mean grouping remains the same.

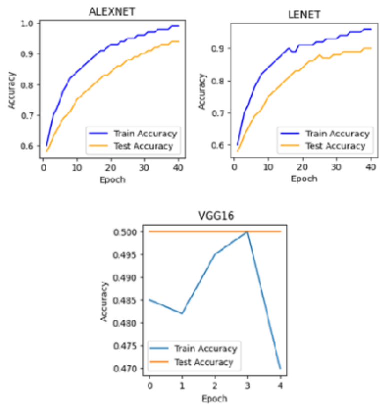


Fig. 5. Accuracy of each model Vs Epoch

Figure 5 and 6 represents the data loss and accuracy of the three DL models in CNN VGG16, Alexnet, and Lenet, at multiple epochs during the training. Each row represents a different epoch, ranging between epoch 10 and 40, with 10 epoch increments. The results include data loss, which measures the error or the difference between the model's prediction and the actual label as regarded from the training dataset. Accuracy refers to the proportion of estimates that are correctly classified: as the training progresses, data loss decreases, which means that the models learn to make better predictions as time goes by. At the same time, accuracy increases as the models get better at classifying which class an instance belongs to. However, each model's performance varies between the same epoch numbers, with some models showing more substantial change. The Alexnet and Lenet accuracies remain high and show constant decrease in data loss at all epochs. While VGG16 demonstrate less performance, it takes more epochs to get to the same accuracy levels.

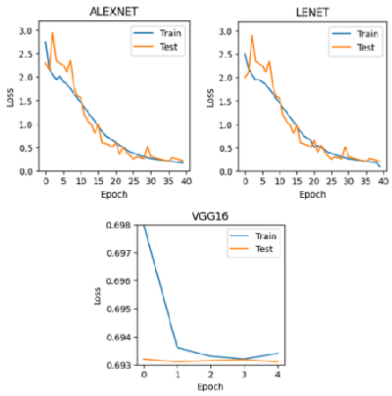


Fig. 6. Data loss of each model vs Epoch

VIII. CONCLUSION

The advanced deep learning models, including CNNs, VGG16, AlexNet, and LeNet exhibits considerable potential for sports shot classification using image datasets. Multiple experiments and evaluations show that these models enable the extraction of distinctive features, thereby improving classification accuracy. CNNs serve as the foundation for hierarchical feature detection, contributing to the ability to differentiate between various types of sports shots. The The VGG16, AlexNet, and LeNet designs use deep convolutional networks tuned for feature extraction to improve classification. VGG16 facilitates the extraction of intricate shot-specific features through its deep architecture and small convolutional filters. AlexNet enhances shot-type detection by introducing effective feature learning with deeper layers and ReLU activations. LeNet, with its simple yet effective structure, enables fundamental pattern recognition, contributing to accurate shot classification by capturing essential spatial hierarchies. Overall, these deep learning models can automate and enhance sports shot classification, contributing to advancements in sports analytics, player performance assessment, and broadcasting technologies.

IX. REFERENCES

[1]. A. Javed, A. Ali Khan, "Shot classification and replay detection for sports video summarization". Front Inform Technol Electron Eng 2022, 23, 790–800, https://doi.org/10.1631/FITEE.2000414.

[2]. I. Bandara, and B. Bačić, "Strokes Classification in Cricket Batting Videos", In Proceedings of 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, CITISIA 2020, pages 1-6.

[3]. R. Kumar, D, Santhadevi Barnabas, Janet. "Outcome Classification in Cricket Using Deep Learning", 2019, IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 55-58. 10.1109/CCEM48484.2019.00012.

[4]. M. Z. Khan, M. A. Hassan, A. Farooq, and M. U. G Khan, "Deep CNN based Data-driven Recognition of Cricket Batting Shots", In Proceedings of International Conference on Applied and Engineering Mathematics, 2018, DOI:10.1109/ICAEM.2018.8536277.

[5]. A. Sen, K. Deb, P. K. Dhar, and T. Koshiba, "CricShotClassify: An Approach to Classifying Batting Shotsfrom Cricket Videos Using a Convolutional Neural.

[6]. Ahmad et al. (2023) introduced an optimized deep learning-based Cricket Activity Focused Network along with a medium-scale benchmark dataset, demonstrating advancements in cricket action recognition (Alexandria Engineering Journal, 73, 771-779). [DOI: 10.1016/j.aej.2023.04.062].

[7]. Foysal et al. (2019) developed Shot-Net, a Convolutional Neural Network (CNN) designed for classifying different cricket shots. This approach was presented at the Second International Conference on Recent Trends in Image Processing and Pattern Recognition (RTIP2R 2018). [DOI: 10.1007/978-981-13-9181-1_10].

[8]. Ahmed et al. (2023) proposed a Multiclass Cricket Shot Classification method using CNN, presented at the 26th International Conference on Computer and Information Technology (ICCIT 2023), highlighting the potential of deep learning in recognizing multiple cricket shots. [DOI: 10.1109/ICCIT60459.2023.10441507].

[9]. Sen et al. (2021) introduced CricShotClassify, an approach integrating Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) for classifying batting shots from cricket videos (Sensors, 21(8), 2846). [DOI: 10.3390/s21082846].

[10]. Gupta and Muthiah (2023) proposed a technique for learning cricket strokes by leveraging spatial and motion visual word sequences, enhancing action classification (Multimedia Tools and Applications, 82(1), 1237-1259). [DOI: 10.1007/s11042-022-13307-y].

[11]. Azhar et al. (2023) introduced a Transformer-based Cricket Shot Classification model, demonstrating the applicability of transformer architectures for cricket action recognition. This study was presented at the International Symposium on Image and Signal Processing and Analysis (ISPA 2023). [DOI: 10.1109/ISPA58351.2023.10279281].

[12]. Devanandan et al. (2021) developed a Random Forest-based approach for cricket shot image classification, offering an alternative machine learning model for sports analytics. This research was presented at the 3rd International Conference on Advancements in Computing (ICAC 2021). [DOI: 10.1109/ICAC54203.2021.9671109].

[13]. Soomro, K., & Zamir, A.R. "Action Recognition in Realistic Sports Videos," in Springer International Publishing, Cham, 2014, pp. 181–208. [Online]. Available: https://doi.org/10.1007/978-3-319-09396-3_9.

[14]. Thomas, G., Gade, R., Moeslund, T.B., Carr, P., & Hilton, A. "Computer Vision for Sports: Current Applications and Research Topics," Computer Vision and Image Understanding, vol. 159, pp. 3–18, 2017. [Online]. Available: https://doi.org/10.1016/j.cviu.2017.04.011, ScienceDirect.

[15]. Wang, J.R., & Parameswaran, N. "Survey of Sports Video Analysis: Research Issues and Applications," in Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing (VIP '05), Australian Computer Society, Inc., Darlinghurst, Australia, 2004, pp. 87–90. [Online]. Available: ACM Digital Library.

[16]. Karmaker, D.; Chowdhury, A.; Miah, M.; Imran, M.; Rahman, M. Cricket shot classification using motion vector. Presented at the 2nd International Conference on Computing Technology and Information Management (ICCTIM), Johor, Malaysia, 21–23 April 2015, pp. 125–129. [CrossRef]

[17]. Khan, A.; Nicholson, J.; Plötz, T. Activity recognition for quality assessment of batting shots in cricket using a hierarchical representation. Proceedings of ACM Interactions on Mobile, Wearable, and Ubiquitous Technologies.

[18]. Semwal, A.; Mishra, D.; Raj, V.; Sharma, J.; Mittal, A. Cricket shot detection from videos. Presented at the 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 10–12 July 2018, pp. 1–6. [CrossRef]

[19]. Harikrishna, N.; Satheesh, S.; Sriram, S.D.; Easwarakumar, K. Temporal classification of events in cricket videos. Presented at the Seventeenth National Conference on Communications (NCC), Bangalore.

[20]. Rafiq, M.; Rafiq, G.; Agyeman, R.; Jin, S.I.; Choi, G.S. Scene classification for sports video summarization using transfer learning. Sensors 2020, 20, 1702. [CrossRef] [PubMed].

Final-Deep Learning-Based Sports Shot Classification System

ORIGINALITY REPORT

7%	4%	6%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.mdpi.com Internet Source	1%
2	Anurag Tiwari, Manuj Darbari. "Emerging Trends in Computer Science and Its Application - Proceedings of the International Conference on Advances in Emerging Trends in Computer Applications (ICAETC-2023) December 21–22, 2023, Lucknow, India", CRC Press, 2025 Publication	1%
3	Ruqiang Yan, Jing Lin. "Equipment Intelligent Operation and Maintenance", CRC Press, 2025 Publication	1%
4	www.ncbi.nlm.nih.gov Internet Source	<1%
5	June Moh Goo, Xenios Milidonis, Alessandro Artusi, Jan Boehm, Carlo Ciliberto. "Hybrid-Segmentor: Hybrid approach for automated fine-grained crack segmentation in civil infrastructure", Automation in Construction, 2025 Publication	<1%
6	coe.panimalar.ac.in Internet Source	<1%
7	Submitted to University of Westminster Student Paper	<1%
8	"Advances in Intelligent Computing Techniques and Applications", Springer Science and Business Media LLC, 2024	<1%

Publication

9	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	<1 %
10	Submitted to City University Student Paper	<1 %
11	Submitted to University of Bolton Student Paper	<1 %
12	"Computational Intelligence in Communications and Business Analytics", Springer Science and Business Media LLC, 2024 Publication	<1 %
13	www.pubfacts.com Internet Source	<1 %
14	Muttena, Sanjivi, S. Sriram, and R. Shiva. "Mapping gestures to speech using the kinect", 2014 International Conference on Science Engineering and Management Research (ICSEMR), 2014. Publication	<1 %
15	ijisae.org Internet Source	<1 %
16	irjaeh.com Internet Source	<1 %
17	1library.org Internet Source	<1 %
18	Mahima Dahiya, Nitin Nitin, Deepak Dahiya. "Intelligent Cyber Security Framework Based on SC-AJSO Feature Selection and HT-RLSTM Attack Detection", Applied Sciences, 2022 Publication	<1 %
19	Faizan Ullah, Abdu Salam, Mohammad Abrar, Farhan Amin. "Brain Tumor Segmentation Using a Patch-Based Convolutional Neural Network: A Big Data Analysis Approach", Mathematics, 2023 Publication	<1 %
20	dokumen.pub Internet Source	<1 %
21	ebin.pub Internet Source	<1 %
22	"Intelligent Data Communication Technologies and Internet of Things", Springer Science and Business Media LLC, 2022 Publication	<1 %

A.5 PAPER PUBLICATION

Publication Name: AIP Conference Conference Details:

- International Conference on Advances in Materials, Manufacturing, and Artificial Intelligence Applications ICAMMAIA 2025.
- Organized By Department Of Mechanical Engineering, Harcourt Butler Technical University Kanpur, Uttar Pradesh, India

Conference Date: 4th – 5th April 2025

NOTIFICATION ACCEPTANCE MAIL RECEIVED:

Acceptance Notification Inbox ×

CONVENER ICAMMAIA 2025

to me

Ref: HBTU/MED/ICAMMAIA_2025_054

Dear Author (s),

I am pleased to inform you that your manuscripts have been accepted for publication in AIP Conference Proceedings.

We have received a payment of Rs. 6500/- from your side.

The technical details will be updated on the website and will be sent to your email.

Thank you for submitting your work to ICAMMAIA 2025.

REFERENCES

REFERENCES

[1]. A. Javed, A. Ali Khan, "Shot classification and replay detection for sports video summarization". Front Inform Technol Electron Eng 2022, 23, 790–800, https://doi.org/10.1631/FITEE.2000414.

[2]. I. Bandara, and B. Bačić, "Strokes Classification in Cricket Batting Videos", In Proceedings of 5th International Conference on Innovative Technologies in Intelligent Systems and Industrial Applications, CITISIA 2020, pages 1-6.

[3]. R. Kumar, D, Santhadevi Barnabas, Janet. "Outcome Classification in Cricket Using Deep Learning", 2019, IEEE International Conference on Cloud Computing in Emerging Markets (CCEM), 55-58. 10.1109/CCEM48484.2019.00012.

[4]. M. Z. Khan, M. A. Hassan, A. Farooq, and M. U. G Khan, "Deep CNN based Data-driven Recognition of Cricket Batting Shots", In Proceedings of International Conference on Applied and Engineering Mathematics, 2018, DOI:10.1109/ICAEM.2018.8536277.

[5]. A. Sen, K. Deb, P. K. Dhar, and T. Koshiba, "CricShotClassify: An Approach to Classifying Batting Shotsfrom Cricket Videos Using a Convolutional Neural.

[6]. Ahmad et al. (2023) introduced an optimized deep learning-based Cricket Activity Focused Network along with a medium-scale benchmark dataset, demonstrating advancements in cricket action recognition (Alexandria Engineering Journal, 73, 771-779). [DOI: 10.1016/j.aej.2023.04.062].

[7]. Foysal et al. (2019) developed Shot-Net, a Convolutional Neural Network (CNN) designed for classifying different cricket shots. This approach was presented at the Second International Conference on Recent Trends in Image Processing and Pattern Recognition (RTIP2R 2018). [DOI: 10.1007/978-981-13-9181-1_10].

[8]. Ahmed et al. (2023) proposed a Multiclass Cricket Shot Classification method using CNN, presented at the 26th International Conference on Computer and

Information Technology (ICCIT 2023), highlighting the potential of deep learning in recognizing multiple cricket shots. [DOI: 10.1109/ICCIT60459.2023.10441507].

[9]. Sen et al. (2021) introduced CricShotClassify, an approach integrating Convolutional Neural Networks (CNNs) and Gated Recurrent Units (GRUs) for classifying batting shots from cricket videos (Sensors, 21(8), 2846). [DOI: 10.3390/s21082

[10]. Gupta and Muthiah (2023) proposed a technique for learning cricket strokes by leveraging spatial and motion visual word sequences, enhancing action classification (Multimedia Tools and Applications, 82(1), 1237-1259). [DOI: 10.1007/s11042-022-13307-y].

[11]. Azhar et al. (2023) introduced a Transformer-based Cricket Shot Classification model, demonstrating the applicability of transformer architectures for cricket action recognition. This study was presented at the International Symposium on Image and Signal Processing and Analysis (ISPA 2023). [DOI: 10.1109/ISPA58351.2023.10279281].

[12]. Devanandan et al. (2021) developed a Random Forest-based approach for cricket shot image classification, offering an alternative machine learning model for sports analytics. This research was presented at the 3rd International Conference on Advancements in Computing (ICAC 2021). [DOI: 10.1109/ICAC54203.2021.9671109].

[13]. Soomro, K., & Zamir, A.R. "Action Recognition in Realistic Sports Videos," in Springer International Publishing, Cham, 2014, pp. 181–208. [Online]. Available: https://doi.org/10.1007/978-3-319-09396-3_9.

[14]. Thomas, G., Gade, R., Moeslund, T.B., Carr, P., & Hilton, A. "Computer Vision for Sports: Current Applications and Research Topics," Computer Vision and Image

Understanding, vol. 159, pp. 3–18, 2017. [Online]. Available: https://doi.org/10.1016/j.cviu.2017.04.011, ScienceDirect.

[15]. Wang, J.R., & Parameswaran, N. "Survey of Sports Video Analysis: Research Issues and Applications," in Proceedings of the Pan-Sydney Area Workshop on Visual Information Processing (VIP '05), Australian Computer Society, Inc., Darlinghurst, Australia, 2004, pp.

[16]. Karmaker, D.; Chowdhury, A.; Miah, M.; Imran, M.; Rahman, M. Cricket shot classification using motion vector. Presented at the 2nd International Conference on Computing Technology and Information Management (ICCTIM), Johor, Malaysia, 21–23 April 2015, pp. 125–129. [CrossRef]

[17]. Khan, A.; Nicholson, J.; Plötz, T. Activity recognition for quality assessment of batting shots in cricket using a hierarchical representation. Proceedings of ACM Interactions on Mobile, Wearable, and Ubiquitous Technologies.

[18]. Semwal, A.; Mishra, D.; Raj, V.; Sharma, J.; Mittal, A. Cricket shot detection from videos. Presented at the 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 10–12 July 2018, pp. 1–6. [CrossRef]

[19]. Harikrishna, N.; Satheesh, S.; Sriram, S.D.; Easwarakumar, K. Temporal classification of events in cricket videos. Presented at the Seventeenth National Conference on Communications (NCC), Bangalore.

[20]. Rafiq, M.; Rafiq, G.; Agyeman, R.; Jin, S.I.; Choi, G.S. Scene classification for sports video summarization using transfer learning. Sensors 2020, 20, 1702. [CrossRef] [PubMed].